Moving mesh methods for numerical solution of porous medium equations

By

Cuong Ngo

Submitted to the graduate degree program in the Department of Mathematics and the
Graduate Faculty of the University of Kansas
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Committee members

Weizhang Huang, Chairperson

Xuemin Tu

Erik Van Vleck

Hongguo Xu

Charlie Zheng

Date defended: April 25, 2017

The Dissertation Committee for Cuong Ngo certifies
that this is the approved version of the following dissertation :

Moving mesh methods for numerical solution of porous medium equations

Weizhang Huang, Chairperson

Date approved: April 25, 2017

Abstract

Porous medium equation (PME) has been found in many applications of the physical sciences. The equation is nonlinear, degenerate, and in many situations has a free boundary, which altogether pose great challenges for mathematical and numerical analyses. In contrast with the mathematical development of PME, which began in the 1950s and has since had much success, studies of numerical solution did not appear until the 1980s. Though a significant progress has been made since then for the 1D setting, only limited success has been observed for 2D cases. In this dissertation, we will propose several moving mesh methods which improve the accuracy and convergence order of the PME numerical solution.

Acknowledgements

I would like to thank the Creator of the universe, the Lord Jesus Christ, who not only made me in His image, but also has given me life abundant, not only in this world, but also in that which is to come. I thank Him for having led me in every detail and aspect toward finishing my secular education.

I would like to express my gratitude toward my dearly beloved parents who have made many great sacrifices to raise me, and also to support my long-term education in the United States. I am deeply thankful for Pastor Chris Matthews and his family, because they have loved and cherished me as a family member, and have supported me spiritually and emotionally for most of my life in Kansas. I am also indebted to my brothers and sisters of Smoky Valley Baptist Church (Lindsborg, KS) and Centropolis Baptist Church (Centropolis, KS, especially Pastor Nelson and his wife Lois) for their Christlike love, care, and hospitality toward me, by whom also I am encouraged to love and serve God with all my heart. I would like to thank the Morains (Glen and Karen), along with many former high school teachers of Canyonville Christian Academy (Canyonville, OR) for making great impacts in my life through their Christ-honoring walks and virtuous lives.

I would like to honor my Uncle and Aunt Ten ("10"), because they not only loved me greatly, but also first gave me the idea of growing up and getting a doctorate. I am especially thankful to Mr. Rosenborg, my former and beloved high school Math teacher and mentor, who in 2004 expressed his belief in God's divine will for me to obtain a PhD in Mathematics, and who has encouraged and inspired me to study well in the discipline ever since. I am greatly indebted to the late Dr. John Murphy,

my beloved professor who had taught me Math for over four years at Bethany College (Lindsborg, KS), and had prepared me much for my graduate study. I am also thankful to Dr. Adebanjo Oriade, my former Physics professor at Bethany College, who diligently took time to teach and inspire me to use Linux operating system, the tool which has proven invaluable to me for my research, learning, and dissertation at the graduate school. I would like to thank Professor Van Tassel and Professor Eric Monder, my former English teachers at Bethany College, for taking great efforts to train me in the great arts of English writing and communication.

I would like to thank my advisor, Professor Weizhang Huang, for accepting me as his student and training me to do research for over five years. Without his vision, guidance, assistance, and support, my thesis would never have succeeded. I would also like to thank Professor Erik Van Vleck, Professor Xuemin Tu, and Professor Hongguo Xu for having various parts in teaching me the great arts of Numerical Analysis and Computational Mathematics, and also many other professors and instructors who have taught me other Mathematical disciplines throughout my graduate school career.

I would like to acknowledge my gratitude toward the staff members and administrators of our Math department, and also of our university (including the International Student Service) for having various parts in helping me, and insuring my well-being and comfort as a student and an employee throughout the past seven years.

The computational aspects of my research for this dissertation were made possible by the Information Technology equipments of the University of Kansas, and also of our Math department. I am particularly indebted to the Advance Computational Facility (ACF) "Supercomputer," which my university invested in 2014, and also to the talented technicians who have worked diligently behind-the-scene, and have assisted me tremendously with the training and usage of the system.

I would like to express my great gratitude to our university and department for

having supported me financially for nearly seven years, not only through entrusting me with a beneficial GTA position, but also through the many summer scholarships, fellowship, grant, and awards throughout my graduate career. I also greatly appreciate each and everyone of my students who have sat under my teaching in the past seven years; without them, there would have been less happiness, excitement, and fulfillment in my journey through the PhD program.

Contents

List of Figures

Chapter 1

Introduction

In this thesis, we seek to solve numerically the porous medium equation (PME)

$$u_t = \nabla \cdot (|u|^m \nabla u), \tag{1.1}$$

from which we pose a Dirichlet initial-boundary value problem (IBVP) in two dimensions,

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u), & \text{in} \quad \Omega \times (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x}), & \text{on} \quad \Omega \\ u(\boldsymbol{x}, t) = 0, & \text{on} \quad \partial\Omega \times (t_0, T] \end{cases} \tag{1.2}$$

where $\Omega \in \mathbb{R}^2$ is a bounded polygonal domain, $u_0(\boldsymbol{x})$ is a given initial solution, and $m \geq 1$ is a physical parameter. PME (1.1) is a nontrivial generalization of the heat equation $u_t = \Delta u$. This equation is found in many areas of the physical sciences, which include, for example, flow of gas in porous medium, incompressible fluid dynamics, nonlinear heat transfer, and image processing; more examples can be found in [46]. When (1.1) is studied in the context of gas flow in porous medium, u represents the density of the gas, $|u|^m$ the pressure, $u\nabla(|u|^m)$ the flux, $\nabla(|u|^m)$ the velocity, and m is the isentropic coefficient. For convenience, we refer to (1.2) as the "U-formulation" for PME.

The equation itself represents a nonlinear diffusion process and possesses many interesting properties. One of the most notable among these is its degeneracy at locations of the domain where the solution is zero, which induces the peculiar "finite propagation" phenomenon (in contrast with the heat equation's "infinite speed of propagation"), meaning, if the initial solution $u_0(\boldsymbol{x})$ has a compact support, the solution will have a compact support for any time $t > t_0$. This in effect creates a free boundary which delineates the zero and nonzero regions of the solution, and propagates with a finite speed for all time. Contrary to the heat equation which smooths out its initial solution, PME solution can become irregular even in cases where it has a smooth initial solution (See Fig. 2.1). Moreover, for some special initial solutions, IBVP (1.2) exhibits the waiting time phenomenon in which the free boundary is stationary initially, but will eventually move after a finite amount of time, called the "waiting time," has elapsed [49]. Developments of mathematical theory for PME began in the 1950s (and some of its generalizations in the late 1980s), and since then much progress has been achieved and exhibited through a large quantity of literature, which includes the earlier work by Oleĭnik et al. [39], Kalašnikov [29], Aronson [2], the more recent work by Shmarev [42, 43], and the monograph by Vázquez [46] and the references therein.

In contrast, numerical studies on PME had a later start, but have attracted much attention from many researchers for the past forty years. The earliest work, to the best knowledge of the author, dates back to 1983 by Rose [40] who, by using a regularization approach and taking the regularization parameter as $\varepsilon = \mathscr{O}(h^{\frac{2m+4}{m^2+4m+2}})$, shows that the error for a P1 finite element (FE) – backward Euler (for time) approximation of PME is bounded by

$$\left(\sum_n \Delta t \|u_h^n - u\|^{m+2}_{L^{m+2}(\Omega)}\right)^{\frac{1}{m+2}} \leq C\left(\Delta t^{\frac{1}{m+1}} + \left(\ln\left(\frac{1}{h}\right)\right)^{\frac{1}{(m+1)(m+2)}} h^{\frac{2}{m+1}}\right), \tag{1.3}$$

where h is the maximum element diameter and u_h^n is the numerical approximation of u at $t = t_n$. Since then, various FE approximations have been developed, and their associated error estimates have been obtained. For example, Nochetto and Verdi [38] study a P1 finite element – 1st order semi-implicit approximation on a class of degenerate parabolic PDEs, and show that the error of

approximation for PME (as its special case) is bounded by

$$\begin{aligned}
&\|u_h-u\|_{L^\infty(0,T;H^{-1}(\Omega))}+\|(u_h)^{m+1}-u^{m+1}\|_{L^2(0,T;L^2(\Omega))}+\left\|\int_0^t((u_h)^{m+1}-u^{m+1})\right\|_{L^\infty(0,T;H^1(\Omega))}\\
&\qquad\le C\left(\frac{h^2}{\varepsilon}+\frac{h^4}{\varepsilon^2\Delta t}+\Delta t\right)^{\frac{1}{2}} \qquad (1.4)\\
&\qquad= \mathscr{O}(h^{\frac{m+2}{2(m+1)}}),\quad \text{if}\quad \Delta t=\mathscr{O}(h^{\frac{m+2}{m+1}}),\quad \varepsilon=\mathscr{O}(h^{\frac{m}{m+1}}),
\end{aligned}$$

which is in fact an improvement on the result of Rose [40]. Besides, error bounds have been found in various norms as well, especially for the P1 finite element – backward Euler approximation on quasi-uniform meshes; some examples of these are

$$\|u_h-u\|_{L^\infty(0,T;H^{-1}(\Omega))}\le C\left(\Delta t+\left(\ln\left(\frac{1}{h}\right)\right)^{\frac{2m+3}{2m+2}}h\right) \qquad (1.5)$$

by Rulla and Walkington [41],

$$\|u_h-u\|_{L^2(0,T;L^2(\Omega))}\le Ch^{\frac{m^2+6m+8}{6m^2+14m+8}}\quad \text{when}\quad \Delta t=\mathscr{O}(h^{\frac{5m+4}{2m}}) \qquad (1.6)$$

by Ebmeyer [16], and

$$\|u_h-u\|_{L^{m+2}(0,T;L^{m+2}(\Omega))}\le C\left(\Delta t^{\frac{1}{2}}+h+h^{\frac{1}{m+1}\left(\frac{dm}{2m+4}+1\right)}\right)^{\frac{1}{m+2}} \qquad (1.7)$$

by Wei and Lefton [47], where d is the space dimension. It can be easily seen from these error estimates that the convergence rate is first order at best and decreases as m gets higher; e.g. [40] shows that the order of convergence diminishes to zero as $m\to\infty$ (cf. Eqn. (1.3)). Further, given what we have known concerning the regularity of PME solution, some of these estimates have been shown to be optimal in their corresponding norms. Besides these, error estimates in quasi-norms have been obtained by Ebmeyer and Liu [17], and more recently, Duque et al. [13] establish $L^{1+\max(\gamma/2)}$ error bounds for the approximation of a general order continuous Galerkin

in space and a general order discontinuous Galerkin in time for PME with a variable exponent $m = \gamma(\boldsymbol{x})$. Moreover, among the approaches for finding these estimates, Emmrich and Šiška [18] have used the theory of monotone operators to prove that a Galerkin finite element – backward Euler approximation converges to the weak solution of PME.

Besides the popular continuous Galerkin FE methods, Zhang and Wu [49] applies a high order local discontinuous Galerkin (LDG) FE method for the one-dimensional PME on a uniform mesh. The major strength of the method is its effectiveness in eliminating spurious oscillations in the numerical solution near the free boundary, by which a higher convergence order is attained within the solution's support and away from the free boundary.

Due to the presence of a free boundary and (in many situations) the low regularity of PME solution (especially one with very steep gradient and corner shape near/at the free boundary), a numerical solution will need more resolution in the mesh in order to produce acceptable results. For the traditional uniform mesh approach, this requires in most situations a very fine mesh to satisfy this need. Typically, usage of a fine mesh is feasible for 1D setting; however, for a larger-scale problem, especially in higher dimensions (such as 2D or 3D), a fine mesh might be very costly in terms of computer memory and processing time, and in many cases impractical. Indeed, this particular challenge in such situations (for which PME is an interesting example) has led to studies of mesh adaptation methods, which began around the 1980s. Typically, there are three types of mesh adaptation. The *h*-method (where "h" is the standard notation which represents the resolution of a spacing discretization, e.g. the size of a mesh element) seeks to improve accuracy by adding more mesh points in regions of the domain where the error in the solution is large. This approach, though intuitive (and in many ways natural), requires a continual increment of mesh points, which in some cases can be forbidding in terms of memory usage. Another approach–the *p*-method (where "p" stands for "polynomials"), seeks to improve the accuracy by successively increasing the order of polynomials in the solution's interpolation. Though the *p*-method improves over the *h*-method in that it does not add more points into the mesh, its implementation in many situations (e.g. with complicated boundary) is much more difficult. On the other hand, the *r*-

method (where "r" stands for "relocation") or the adaptive moving mesh method (or the moving mesh method in short) maintains the mesh structure (i.e. not introducing more mesh points or elements into the mesh) throughout the process, and seeks to move the mesh points to regions of the solution's domain that require more definition. Over the years, a number of moving mesh methods have been developed; among these are the ALE (Arbitrary Lagrangian-Eulerian) method by Hirt et al. [21], the MFE (moving finite element) method by Miller and Miller [36], the deformation map method by Liao and Anderson [35], the mesh-rezoning method by Li et al. [33], the method based on geometric conservation law by Cao et al. [11], and the method based on the conservation of fraction mass by Baines et al. [5]. More information can be found in [3, 6, 8, 10, 26, 45] along with the references therein. Moreover, in some sophisticated applications, the r-method can also be combined with other adaptation techniques (e.g. the h-method and the p-method) to produce greater accuracy and robustness.

Intuitively, an adaptive moving mesh method is more desirable, and apparently a more natural choice for a free-boundary problem such as PME, since the moving mesh can follow free boundaries dynamically, and adaptivity has potential in improving accuracy, especially for a solution having low regularity and/or steep front. Indeed, mesh adaptation methods for PME have gained more attention and active studies from many researchers since the late 1990s. For example, in 1999, Budd et al. [9] apply the Moving Mesh PDE (MMPDE) moving mesh method [25, 26] for finding self-similarity solutions of one-dimensional PME, where a special monitor function is designed for preserving the scaling invariance of PME. In a 2005-2006 series of papers [4, 5, 7] (also see the review paper [6]), Baines et al. develop a moving mesh FE method for PME in one and two dimensions, where the movement of mesh points is based upon conserving a local proportion of the total mass that is present in the projected initial data. In simulating the standard/benchmark Barenblatt-Pattle (BP) special solutions, the method achieves a second order convergence when applied to (1.2) for $m = 1$ for both the 1D and 2D settings [5, 6]. However, for cases of $m > 1$ (with a representative example of $m = 3$ in [5]), this method only yields a first order convergence in 1D. Nevertheless, a second order convergence can be recovered if an optimal initial mesh, com-

puted via a special algorithm, is used. Unfortunately, Baines et al. have not yet produced such an optimal mesh for the same problem in 2D, since the mentioned algorithm is significantly more computationally expensive for higher dimensional settings. Recently, Duque et al. [14, 15] applies the MMPDE moving mesh FE method for PME with variable exponents with/without absorption. However, when tested for a class of special solutions, the method only yields a first-order convergence.

In this dissertation, an adaptive moving mesh finite element method is developed and studied for the numerical solution of PME. Similar to some previous works such as [14, 15], our method will be based on an MMPDE; but unlike these, it has a few significant improvements. First, our mesh generation method is concerned with minimizing an energy functional (cf. (2.42)) based on the equidistribution principle and alignment condition (cf. Eqn. (2.40) and (2.41)). The equidistribution principle in the moving mesh context essentially means a relocation of the mesh points, so as to make a quantity of interest (such as the interpolation error) constant on each mesh element. While the equidistribution concept was developed in the 1970s and exhibited in the works of de Boor and Burchard (See [48] for references), the alignment condition has been recently developed by Huang [22]. Secondly, unlike [14, 15] where the mesh adaptivity is controlled by a scalar function, the mesh generation and adaptation of our method are controlled by a matrix-valued function, called a metric tensor. The usage of such tensors is more advantageous since it holds information about the solution which effectively directs the MMPDE method to not only control the size of mesh elements, but also their shape and orientation. This in effect causes the mesh to align more naturally with the geometry of the physical solution, whereas the method using a scalar mesh adaptation function does not have such an advantage. In this work, we consider three basic meshing strategies, with an additional one to be mentioned. The first one is nonadaptive and uses a uniform mesh. The second, called arclength-based adaptivity, uses an adaptive mesh based on minimizing the error estimate derived from a piecewise constant interpolation. The third, called Hessian-based adaptivity, seeks to move the mesh points in order to minimize the error estimate derived from a piecewise linear interpolation. Our moving mesh method achieves a breakthrough

with the Hessian-based adaptivity, especially in dealing with the famous Barenblatt-Pattle solutions (cf. (2.14)), notorious for their lack of regularity at the free boundary when $m > 1$.

Generally speaking, when dealing with free/moving boundary problems, there are two major approaches. The first, which can be called "immersed-boundary" or "embedding" approach, performs computations on a fixed (usually rectangular) domain larger than the support of the solution at all time (cf. Fig. 2.5a). Though more memory and more CPU time are required for having extra mesh points outside the support of the solution, and though the solution unavoidably has a lower regularity due to its extension to the larger domain, this approach is very robust for handling more complicated structures in the solution's geometry and in the differential equation itself, since we do not need to explicitly trace the free boundary using Darcy's law. The second strategy, which we term "nonimmersed-boundary" or "nonembedding approach", discretizes only within the support of solution (cf. Fig. 2.5b), where at each new time level, the boundary points have to be moved manually via Darcy's law, and the mesh points then need to be redistributed over the support. Though not having the robustness in dealing with more complex geometric structures in some solutions, and though the accuracy is highly dependent on the precision of Darcy's law computation, the approach itself is advantageous in terms of memory economy and accuracy. The reason for the former is obvious (as we need not to have extraneous mesh points), but for the latter, it is because when considered only within its support, the solution has a higher regularity than it does when considered on a larger domain containing its free boundary. Additionally, for this nonembedding approach, besides the arclength-based and Hessian-based mesh adaptations, we also consider one that specifically places more mesh points at the free boundary; we will term this the "boundary-based" adaptivity.

The outline of the work is as follow. Chapter 1 gives an introduction to the thesis, highlights some significant contributions, and presents some major ideas. Chapter 2 presents some background materials which include the derivation of PME, the well-posedness of (1.2), and some helpful facts and properties regarding the Barenblatt-Pattle solutions. Besides, we will summarize some significant contributions to the numerical solution for PME. In the last part of the chapter,

we will describe the moving mesh finite element method which includes the linear finite element discretization of a PDE on a moving mesh, and the generation of an adaptive moving mesh based on the MMPDE approach (for both the embedding and nonembedding approaches). In the subsequent three chapters, we will present three methods for PME simulation, along with their supporting numerical experiments. Chapter 3 presents our first embedding approach, called the UE-method, where "U" stands for the original formulation in the u-variable–cf. (1.2), and "E" stands for "embedding." In the numerical section of the chapter, besides testing the method against the Barenblatt-Pattle solution for accuracy, we will also illustrate its robustness via problems which have solutions with, for example, complex supports and waiting time phenomenon. Additionally, the UE-method's flexibility is demonstrated in more generalized equations of PME, including those with absorption terms and variable exponents. The second method is nonembedding and based on the U-formulation, and is presented in Chapter 4. Chapter 5 presents a very similar numerical method as in Chapter 4, but it seeks the PME solution indirectly via its "Mathematician's pressure" $v := \frac{|u|^m}{m}$ through a modified statement of the original IBVP (1.2)–cf. (5.5), called the "V-formulation." The same chapter ends with a discussion on computing PME via the V-formulation by the embedding approach. Lastly, Chapter 6 gives a conclusion and also comments for future research.

Chapter 2

Background

In this chapter, we present a brief summary of the porous medium equation (PME) regarding its origin (including its derivation and historical motivation), applications (both classical and modern), and Mathematical properties. We will also introduce the Barenblatt-Pattle solutions, a class of special solutions for PME frequently used for benchmarking numerical methods in literature, and investigate some regularity properties of these solutions, which are relevant for some numerical observations in the subsequent chapters. Some significant contributions to the numerical solution of PME are highlighted (in chronological order) in Section 2.4, and Section 2.5 presents our numerical method.

2.1 Applications of the porous medium equation

The PME

$$u_t = \nabla \cdot (|u|^m \nabla u), \qquad m \geq 1, \tag{2.1}$$

is a nonlinear and degenerate parabolic partial differential equation, whose applications are found in many fields of the mathematical and physical sciences. The following applications, which are most typical for this type of equation, are referenced from the recent monograph by Vázquez [46].

1. *Gas flow through a porous medium.*

   Movements of an ideal gas in a homogeneous porous medium can be modeled in terms of its density, pressure, and velocity, and are solely based on three relationships among these quantities, namely (1) the mass balance equation, (2) Darcy's law, and (3) the equation of state [46]. The model has the form

$$\frac{\partial \rho}{\partial t} = c\Delta(\rho^m),$$

   where c is a constant depending on the three physical quantities mentioned. The equation can be cast into the form of the standard PME in (2.1).

2. *Nonlinear heat transfer.*

   An application which is of historical significance for the development of PME is the theory of heat propagation, where the thermal conductivity depends on the temperature. The governing model, in the absence of heat source and sink, generally has the form

$$c\rho\frac{\partial T}{\partial t} = \nabla\cdot(\kappa\nabla T), \tag{2.2}$$

   where T is the temperature, c is the specific heat, ρ is the density of the medium, and κ is the thermal conductivity. Equation (2.2) is a more general form of the standard PME in (2.1).

3. *Groundwater flow and Boussinesq's equation.*

   A problem arises in fluid dynamics where we need to model the flow of an incompressible fluid through a porous layer. Under some simplification of the hypotheses, the boundary of the fluid's occupied region can be modeled by Boussinesq's equation

$$\frac{\partial h}{\partial t} = \kappa\Delta(h^2),$$

   where h represents the flow's boundary, κ depends on the gravity constant, the permeability

of the medium, and the density and viscosity of the fluid. Again, this equation can be cast into the standard form of PME in (2.1).

4. *Population dynamics.* In biology, the population dynamics of a single species can be represented by the following equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (\kappa \nabla u) + f(u), \tag{2.3}$$

where u models the density of the population, $f(u)$ specifies an interaction within the species, and κ is a diffusivity which depends on the density. As indicated in [46, pg. 25], a reasonable assumption on this diffusivity is that

$$\kappa = \phi(u),$$

where $\phi(\cdot)$ is increasing. This is due to the tendency of the species to avoid crowding ([46]). Under the assumption that $\kappa = \phi(u) = Su$ (where S is a constant) and that $f(u) = 0$, (2.3) becomes

$$\frac{\partial u}{\partial t} = \nabla \cdot (Su \nabla u).$$

This equation again can be cast into the standard form (2.1) of the PME.

5. *Other applications.*

PME can also be found in other areas of science and engineering, such as:

- Thin liquid film spreading under gravity.
- Unsaturated filtration.
- Immiscible fluids. Oil equations.
- Boundary layer theory.
- Spread of magma in volcanoes.
- Limits of kinetic and radiation models.

- Limit of particle models.
- Diffusive coagulation-fragmentation models.
- Diffusion in semiconductors.
- Image processing.
- Some stochastic models.

The interested reader is referred to Vázquez's monograph [46] for more information.

2.2 The mathematical theory of PME

2.2.1 Derivation of the Porous Medium Equation

The flow of an ideal gas through a porous medium involves its density ρ, its pressure p, and its velocity $\boldsymbol{V}$, and is governed by the following three empirical laws (with notation adapted from [46]).

1. Mass balance:

$$\varepsilon \rho_t + \nabla \cdot (\rho \boldsymbol{V}) = 0\,, \tag{2.4}$$

where $\varepsilon \in (0,1)$ is the porosity of the medium.

2. Darcy's law:

$$\mu \boldsymbol{V} = -k \nabla p\,, \tag{2.5}$$

where $\mu > 0$ and $k > 0$ are the viscosity of the fluid and the permeability of the medium, respectively.

3. The equation of state:

$$p = p_0\, \rho^{\gamma}\,, \tag{2.6}$$

where $p_0 > 0$ is a reference pressure, and γ is the polytropic exponent.

Now, assume that ε, μ, k, and p_0 are all constant, we can combine (2.5) and (2.6) and have

$$\boldsymbol{V} = -\frac{k}{\mu}\nabla(p_0\,\rho^\gamma)\,.$$

By using this identity, (2.4) will become

$$\begin{aligned}
\varepsilon\rho_t + \nabla\cdot\left(-\frac{\rho k}{\mu}\nabla(p_0\,\rho^\gamma)\right) &= 0\,,\\
\rho_t + \nabla\cdot\left(-\frac{\rho\, p_0 k}{\varepsilon\mu}\nabla(\rho^\gamma)\right) &= 0\,,\\
\rho_t - \frac{p_0 k}{\varepsilon\mu}\nabla\cdot(\rho\nabla(\rho^\gamma)) &= 0\,,\\
\rho_t - \frac{p_0 k}{\varepsilon\mu}\nabla\cdot\left(\rho\gamma\rho^{\gamma-1}\nabla\rho\right) &= 0\,,\\
\rho_t - \frac{p_0\gamma k}{\varepsilon\mu}\nabla\cdot(\rho^\gamma\nabla\rho) &= 0\,.
\end{aligned}$$

Hence, we have

$$\rho_t = c\nabla\cdot(\rho^m\nabla\rho)\,, \tag{2.7}$$

where

$$c = \frac{p_0\gamma k}{\varepsilon\mu} \quad \text{and} \quad m = \gamma.$$

Moreover, by a time rescaling $\tilde{t} = ct$, we can rewrite (2.7) into

$$\rho_{\tilde{t}} = \nabla\cdot(\rho^m\nabla\rho)\,,$$

from which we have a standard form of the porous medium equation (2.1).

2.2.2 Weak solution and properties

Given a bounded domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, which has a Lipschitz continuous boundary $\Gamma := \partial\Omega$, we consider the homogeneous Dirichlet initial-boundary value problem

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u), & \text{in} \quad \Omega \times (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x}), & \text{in} \quad \Omega \\ u(\boldsymbol{x}, t) = 0, & \text{on} \quad \partial\Omega \times (t_0, T], \end{cases} \tag{2.8}$$

where the initial solution $u_0 \in L^1(\Omega)$. We would like to define a class of weak solutions for problem (2.8). The following definition (adapted from [46, pg. 86]) gives a weak formulation to (2.8) and additionally takes into account the initial condition.

Definition 2.2.1. A locally integrable function u defined in $\Omega \times (t_0, T]$ is said to be a weak solution of (2.8) if

1. $u \in L^1(\Omega \times (t_0, T))$ and $u^{m+1} \in L^1(t_0, T : W_0^{1,1}(\Omega))$; and

2. u satisfies the identity

$$\iint_{\Omega \times (t_0, T)} (|u|^m \nabla u \cdot \nabla \varphi - u \varphi_t) \, d\boldsymbol{x} dt = \int_\Omega u_0(\boldsymbol{x}) \varphi(\boldsymbol{x}, t_0) \, d\boldsymbol{x} \tag{2.9}$$

for each test function $\varphi \in C^1(\overline{\Omega} \times [t_0, T])$, where $\varphi|_{\partial\Omega} = 0$ and $\varphi(\cdot, T) = 0$.

The following theorem [46, pg. 88] gives a sufficient condition for a function $u(\boldsymbol{x}, t)$ to be a weak solution of problem (2.8), which features an intuitive limiting of the solution to u_0 as $t \to 0$. It can also be used as an alternative definition for a weak solution.

Theorem 2.2.1. *Let $u \in L^1(\Omega \times (t_0, T))$ be such that*

1. $u^{m+1} \in L^1(t_0, T : W_0^{1,1}(\Omega))$;

2. *u satisfies the identity*

$$\iint_{\Omega\times(t_0,T)} \left(|u|^m \nabla u \cdot \nabla \varphi - u\varphi_t\right) d\boldsymbol{x}dt = 0 \tag{2.10}$$

for each test function $\varphi \in C_c^\infty(\Omega \times (t_0,T))$*; and*

3. $u(t) \in L^1(\Omega)$ *for each* $t > 0$*, and* $u(t) \to u_0$ *as* $t \to t_0$ *in* $L^1(\Omega)$*,*

then, u is a weak solution to (2.8) *according to Definition 2.2.1.*

In literature, the results for existence and uniqueness of the weak solution to Problem (2.8) are known. Here, we present a version of these results adapted from [46].

The following theorem, found in [46, pg. 93], says that a weak solution to (2.8) exists under some mild analytical assumptions on u_0.

Theorem 2.2.2 (**Existence**). *Suppose the initial solution* $u_0 \in L^1(\Omega)$*,* $u_0^{m+2} \in L^1(\Omega)$*, and* $u_0 \geq 0$*, then there exists a nonnegative weak solution to* (2.8) *(according to Definition 2.2.1) on the time interval* (t_0,∞)*. Moreover, we have* $u^{m+2} \in L^\infty(t_0,T : L^1(\Omega))$ *for all* $T > 0$*,* $u^{m+1} \in L^2(t_0,T : H_0^1(\Omega))$*, and also the energy estimate*

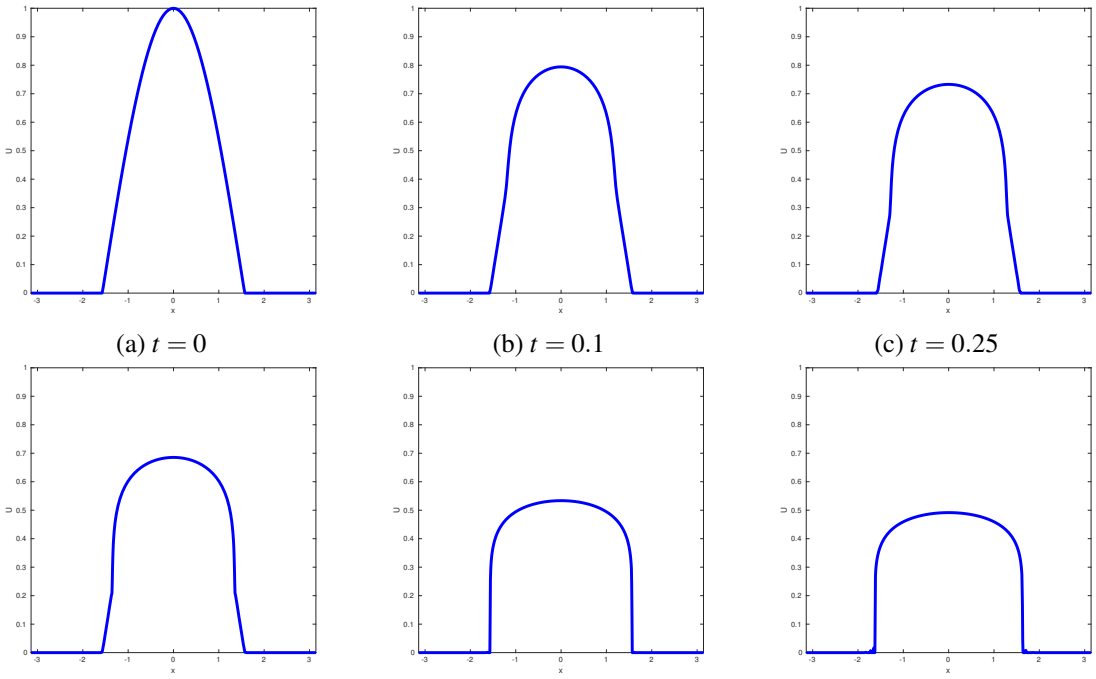
$$\iint_{\Omega\times(t_0,T)} \left||u|^m \nabla u\right|^2 d\boldsymbol{x}dt + \int_\Omega \frac{u^{m+2}(\boldsymbol{x},T)}{m+2}\, d\boldsymbol{x} \leq \int_\Omega \frac{u_0^{m+2}(\boldsymbol{x})}{m+2}\, d\boldsymbol{x}\,. \tag{2.11}$$

Furthermore, the weak solution satisfies the Comparison Principle: If $u, \hat{u}$ *are weak solutions with initial data such that* $u_0 \leq \hat{u}_0$ *a.e. in* Ω*, then* $u \leq \hat{u}$ *a.e in* $\Omega \times (t_0,\infty)$*. In particular if* $u_0 \geq 0$ *in* Ω*, then* $u \geq 0$ *in* $\Omega \times (t_0,\infty)$*.*

The following theorem says that a weak solution to (2.8), if it exists, is unique under some further analytical assumptions on the solution [46, pg. 90].

Theorem 2.2.3 (**Uniqueness**). *Given a weak solution u to* (2.8)*, if we assume further that* $u^{m+1} \in L^2(t_0,T : H_0^1(\Omega))$ *and* $u \in L^2(\Omega \times (t_0,T))$*, problem* (2.8) *has at most one weak solution.*

Moreover, it is also known that PME solution can lose its regularity as time evolves. Fig. 2.1 gives an illustration, where the PME initial solution is smooth, but it gradually becomes irregular to the point of developing very steep/infinite slope near/at the free boundary. More detailed discussion on the regularity of PME solution can be found in [46].



(a) $t = 0$ (b) $t = 0.1$ (c) $t = 0.25$

(d) $t = 0.5$ (e) $t = 5$ (f) $t = 10$

Figure 2.1: Loss of regularity in a PME solution as time evolves.

2.2.3 Propagation properties

In this subsection we would like to present some facts regarding the propagation of the support of a PME solution. Unlike the classic heat equation which has the infinite speed of propagation, the degenerate nature of the PME results in the finite propagation property and has attracted much attention from researchers since the 1950s. Before stating the results relevant to our research, for convenience, we will adopt the following notation from [46]. Let u be a continuous solution defined in Ω_T. We denote the *positivity set* of u as $\mathscr{P}_u$ and define it as the subset of Ω_T whereon u is positive, i.e.

$$\mathscr{P}_u := \{(\boldsymbol{x},t) \in \Omega_T : u(\boldsymbol{x},t) > 0\}.$$

Additionally, we define

$$\mathscr{P}_u(t) := \{\boldsymbol{x} \in \Omega : u(\boldsymbol{x},t) > 0\}$$

as the positivity set only at time level t. Quite naturally, we can also define the support of u at time t, $\mathscr{S}_u(t)$, as the closure of $\mathscr{P}_u(t)$ in Ω, i.e.

$$\mathscr{S}_u(t) := \overline{\mathscr{P}_u(t)}^{\Omega}.$$

It is known that the positivity set of a PME solution is noncontracting ([46, pg. 335]), that is, for every $t_0 < t_1 < t_2$, $\mathscr{P}_u(t_1) \subset \mathscr{P}_u(t_2)$. In fact, it is ever expanding, as we can see from the following theorem [46, pg. 336].

Theorem 2.2.4. *Let u be a nontrivial local solution of the PME defined in $\Omega \times (t_0, \infty)$, where Ω is connected. Then, every point of Ω is absorbed in finite time by the positivity set of u, i.e.*

$$\bigcup_{t>0} \mathscr{P}_u(t) \supseteq \Omega.$$

Moreover, any compact subset K of Ω is covered by $\mathscr{P}_u(T)$ for some finite time T which depends on m, d, the initial data, the geometry of Ω, and the distance between K and $\partial\Omega$.

Notice that Theorem 2.2.4 also implies that $\mathscr{S}_u(t)$ will cover $\overline{\Omega}$ in finite time. Moreover, PME solution u has the uniform finite propagation property, meaning that for $t_0 \le t_1 \le t_2$, $\mathscr{S}_u(t_2)$ belongs to a neighborhood of $D(|t_2 - t_1|)$ of $\mathscr{S}_u(t_1)$, where D is a continuous function $\mathbb{R}_+ \mapsto \mathbb{R}_+$ with $D(0+) = 0$, and D is also independent of the solution u [46, pg. 337-340]. Further, $\mathscr{S}_u(t)$ expands continuously in time, i.e. if Ω is bounded there exist uniform constants δ and $C > 0$ such that for every $0 < h < \delta$, $\mathscr{S}_u(t+h)$ is included in the neighborhood of radius $Ch^{1/2}$ of $\mathscr{S}_u(t)$, namely

$$\mathscr{S}(t+h) \subset \mathscr{S}(t) + B_{Ch^{1/2}}(\mathbf{0}).$$

Consequently, the finite propagation property induces the free boundary $\Gamma(t) := \partial\mathscr{S}_u(t)$, and the

existence of which is formally stated in the following theorem [46] for a more general situation.

Theorem 2.2.5. *Let u be a continuous and bounded strong solution of the PME defined in a space-time cylinder $\Omega \times [t_0, T)$ and assume that $u(\boldsymbol{x}, t_0)$ vanishes in a ball $B \subset \Omega$. Then, the free boundary is a non-empty set.*

In [42], Shmarev studies the Cauchy problem for the nonlinear degenerate parabolic equation (in dimensions $d = 1, 2$, and 3) in the form

$$u_t = \Delta(|u|^m) - au^p \quad \text{in} \quad \mathbb{R}^d \times (t_0, T],$$

where $m > 1$, $p > 0$, $a \in \mathbb{R}$, and $m + p \geq 2$. The result is relevant to the porous medium equation in our case, when parameter $a = 0$. In particular, it is shown that for each time $t \in (t_0, T)$, the free boundary $\Gamma(t) := \partial \mathscr{S}_u(t)$ has the velocity

$$\Gamma'(t) = \lim_{\boldsymbol{x} \to \Gamma(t)-} \left(-\frac{m}{m-1} \nabla(u^{m-1}) + \nabla \Pi \right) \cdot \hat{\boldsymbol{n}}, \tag{2.12}$$

where $\hat{\boldsymbol{n}}$ denotes the unit outward normal to $\Gamma(t)$ and $\Pi(\boldsymbol{x}, t)$ is a solution of the degenerate elliptic equation

$$\nabla \cdot (u \nabla \Pi) = au^p, \quad \Pi = 0 \text{ on } \Gamma(t).$$

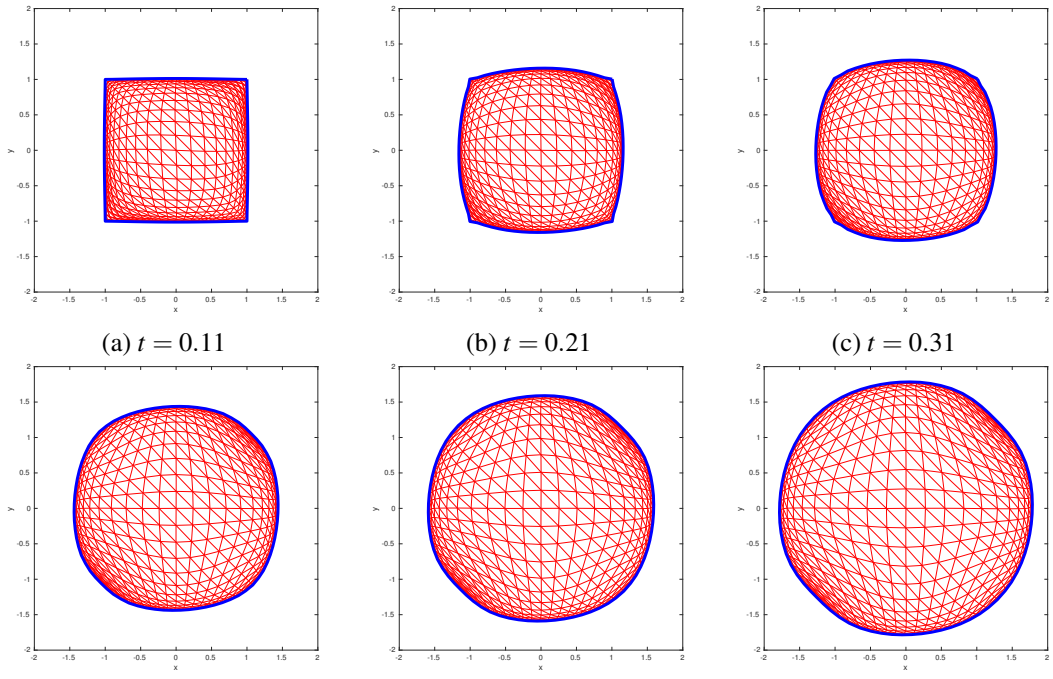
Formula (2.12) indeed is a generalized form of the classic Darcy's law. For PME expressed in the form of equation (2.8), the speed of the boundary is given by

$$\Gamma'(t) = \lim_{\boldsymbol{x} \to \Gamma(t)-} -\nabla \left(\frac{|u|^m}{m} \right) \cdot \hat{\boldsymbol{n}}, \tag{2.13}$$

which is the form of Darcy's law to be used with the U-formulation of PME for the rest of the dissertation. In addition, one of the fascinating phenomena of PME is the existence of a waiting time for a certain type of initial solutions, where the free boundary does not move until a finite amount of time has elapsed (See Examples 3.3.3, 4.5.2, and 5.5.2). Loosely speaking, from Darcy's

law given in (2.13), this phenomenon could be expected for initial solutions having vanishing $\nabla(|u|^m)$ at the initial free boundary.

Another phenomenon worthy of our consideration is the high regularity of the free boundary. While the solution of PME can lose its regularity, its free boundary always becomes smoother. In fact, according to the work by Daskalopoulos and Hamilton [12], the free boundary enjoys C^∞ regularity on (t_0, T), i.e. instantly after the initial time t_0. Fig. 2.2 shows a simulation of the free boundary of a PME solution, which initially encloses a square support. Notice that the free boundary is turned into a circular shape after a finite time.



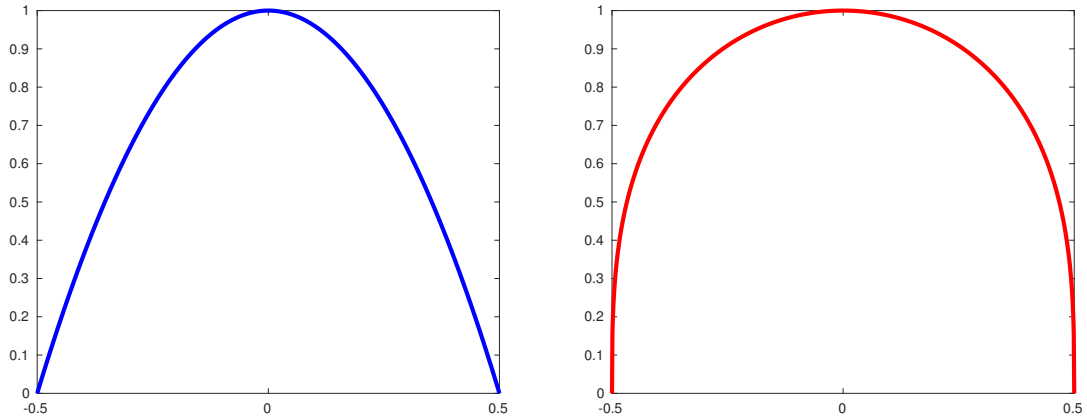
(a) $t = 0.11$ (b) $t = 0.21$ (c) $t = 0.31$

(d) $t = 0.51$ (e) $t = 0.76$ (f) $t = 1.2$

Figure 2.2: The smoothing effect on the free boundary of PME solution as time evolves.

2.3 The Barenblatt-Pattle solution to the PME

A few classes of special solutions to PME have been discovered over the years; among these are the Barenblatt-Pattle solutions. This special class of self-similar solutions has been widely used by researchers to evaluate the accuracy of their numerical methods for PME over the past few decades.



(a) 1D | $m = 1$ (b) 1D | $m > 1$

Figure 2.3: Typical Barenblatt-Pattle solutions 1D.

These solutions have the form

$$u(r,t) = \begin{cases} \frac{1}{\lambda^d(t)} \left(1 - \left(\frac{r}{r_0\lambda(t)}\right)^2\right)^{\frac{1}{m}}, & \text{for} \quad |r| \le r_0\lambda(t) \\ 0, & \text{for} \quad |r| > r_0\lambda(t) \end{cases} \tag{2.14}$$

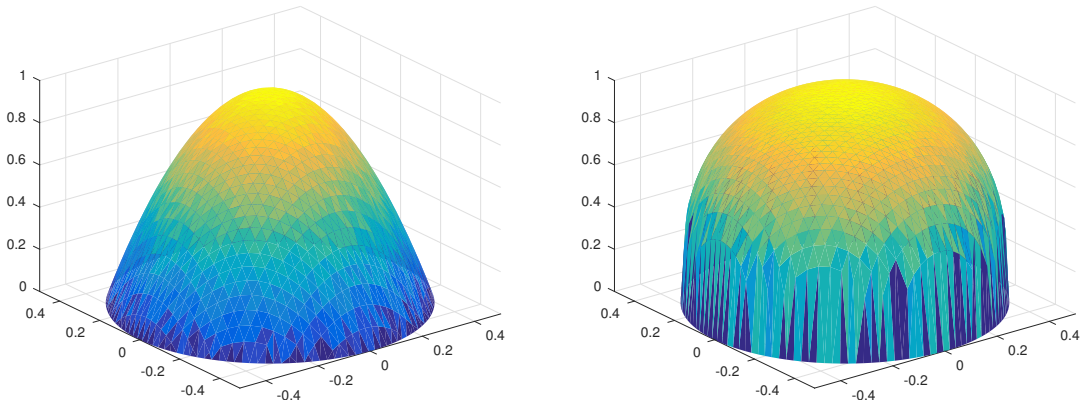
where

$$m \ge 1\,, \quad r = |\boldsymbol{x}|, \quad \lambda(t) = \left(\frac{t}{t_0}\right)^{\frac{1}{2+dm}}, \quad t_0 = \frac{r_0^2 m}{2(2+dm)}, \tag{2.15}$$

and $r_0 > 0$ is a given parameter. These solutions have some interesting characteristics; the most notable among these is their regularity at/near the boundary. For the case of $m = 1$, the slope of the BP solution is finite near the free boundary (see Fig. 2.3a, 2.4a). However, for cases where $m > 1$, the slope of these solutions at/near the free boundary is very steep or infinite (see Fig. 2.3b, 2.4b) and the regularity decreases as m increases; this causes great challenges for numerical simulation of PME.

The following theorems offer some insights to the regularity of the Barenblatt-Pattle solutions. They will be referred to later in Chapter 3, in connection with the convergence behavior of our first moving mesh method.

Theorem 2.3.1. *The Barenblatt-Pattle solution u in* (2.14) *belongs to $H^1(supp(u))$ if $m = 1$.*



(a) 2D | $m = 1$ (b) 2D | $m > 1$

Figure 2.4: Typical Barenblatt-Pattle solutions 2D.

Moreover, for $m > 1$, $u \in W^{1,\frac{m}{m-1}-\varepsilon}(supp(u))$, *for some* $\varepsilon > 0$.

Proof. We observe that for each time t and position $\boldsymbol{x}$ in the support of u, the BP solution takes the form

$$u(r) = A\left(1 - Br^2\right)^{\frac{1}{m}}.$$

Without loss of generality, we consider the simplified situation where $A = B = 1$, i.e.

$$u(r) = \left(1 - r^2\right)^{\frac{1}{m}}, \quad \text{for} \quad 0 \le r \le 1.$$

We observe also that near $r = 1$,

$$u(r) \approx (1 - r)^{\frac{1}{m}}, \quad \text{for} \quad r \approx 1.$$

Now, in the polar coordinate system, we have

$$\begin{aligned} r &= \sqrt{x^2 + y^2}, \qquad \frac{\partial r}{\partial x} = \frac{x}{r}, \\ \frac{\partial u}{\partial x} &= \frac{\partial u}{\partial r}\frac{\partial r}{\partial x} = \frac{\partial u}{\partial r}\frac{x}{r} = \frac{x}{r}\frac{\partial u}{\partial r}, \quad \text{and} \\ \frac{\partial u}{\partial y} &= \frac{y}{r}\frac{\partial u}{\partial r}. \end{aligned} \tag{2.16}$$

Now, for $m = 1$, $u(r) = 1 - r^2$ and we have

$$\begin{aligned}\int_{\text{supp}(u)} |\nabla u|^2 \, dxdy &= \int_{\text{supp}(u)} (u_x^2 + u_y^2) \, dxdy = \int_0^{2\pi} \int_0^1 \left(\frac{\partial u}{\partial r}\right)^2 \, r dr d\theta \\ &= \int_0^{2\pi} \int_0^1 (-2r)^2 \, r dr d\theta < \infty,\end{aligned}$$

which implies that $u \in H^1(\text{supp}(u))$. For the case $m > 1$, we consider

$$\begin{aligned}I = \int_{\text{supp}(u)} |\nabla u|^\alpha \, dxdy &= \int_{\text{supp}(u)} (|\nabla u|^2)^{\frac{\alpha}{2}} \, dxdy = \int_{\text{supp}(u)} (u_x^2 + u_y^2)^{\frac{\alpha}{2}} \, dxdy \\ &= \int_0^{2\pi} \int_0^1 \left[\left(\frac{\partial u}{\partial r}\right)^2\right]^{\frac{\alpha}{2}} \, r dr d\theta \\ &= \int_0^{2\pi} \int_0^1 \left(\frac{1}{m^2}\right)^{\frac{\alpha}{2}} (1-r)^{\left(\frac{1}{m}-1\right)\alpha} \, r dr d\theta \,.\end{aligned}$$

In order for $I < \infty$, we must have

$$\left(\frac{1}{m} - 1\right)\alpha + 1 > 0 \quad \Longrightarrow \quad \alpha < \frac{m}{m-1} \,.$$

Hence, $u \in W^{1, \frac{m}{m-1} - \varepsilon}(\text{supp}(u))$, for some $\varepsilon > 0$, as concluded. □

Theorem 2.3.2. *If u is a Barenblatt-Pattle solution to PME given in equation* (2.14)*, then $\sqrt{\det(H(u))} \in L^{\frac{2}{3}}(\text{supp}(u))$, where $H(u)$ is the Hessian of u.*

Proof. Similar to the proof of Theorem 2.3.1, we consider the simplified situation where

$$u(r) = \left(1 - r^2\right)^{\frac{1}{m}}, \quad \text{for} \quad 0 \le r \le 1 \,,$$

and

$$u(r) \approx (1-r)^{\frac{1}{m}}, \quad \text{for} \quad r \approx 1 \,.$$

We would like to consider $\sqrt{\det(H(u))}$, where

$$H(u) = \begin{bmatrix} u_{xx} & u_{xy} \\ u_{xy} & u_{yy} \end{bmatrix}.$$

For this, we compute

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{x}{r}\frac{\partial u}{\partial r}\right) \qquad \text{(cf. (2.16))} \\ &= \frac{\partial}{\partial x}\left(x\frac{1}{r}\frac{\partial u}{\partial r}\right) = 1\left(\frac{1}{r}\frac{\partial u}{\partial r}\right) + x\frac{\partial}{\partial x}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right) = \frac{1}{r}\frac{\partial u}{\partial r} + x\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right)\frac{\partial r}{\partial x}\right] \\ &= \frac{1}{r}\frac{\partial u}{\partial r} + \frac{x^2}{r}\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right)\right]. \end{aligned}$$

Similarly, we have

$$\begin{aligned} \frac{\partial^2 u}{\partial y^2} &= \frac{1}{r}\frac{\partial u}{\partial r} + \frac{y^2}{r}\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right)\right], \\ \frac{\partial^2 u}{\partial x \partial y} &= \frac{xy}{r}\left[\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right)\right]. \end{aligned}$$

By direct computation, we have

$$\sqrt{\det(H(u))} = \sqrt{\frac{1}{r}\frac{\partial u}{\partial r}\frac{\partial^2 u}{\partial r^2}} \approx \sqrt{\frac{\partial u}{\partial r}\frac{\partial^2 u}{\partial r^2}} \quad \text{for} \quad r \approx 1.$$

Since

$$\frac{\partial u}{\partial r} \approx (1-r)^{\frac{1}{m}-1} \quad \text{and} \quad \frac{\partial^2 u}{\partial r^2} \approx (1-r)^{\frac{1}{m}-2},$$

we further have

$$\sqrt{\det(H(u))} \approx (1-r)^{\frac{1}{m}-\frac{3}{2}}.$$

Now we consider the integral

$$I = \int_{\mathrm{supp}(u)} \left[\sqrt{\det(H(u))}\right]^{\alpha} dxdy \approx \int_0^{2\pi}\int_0^1 (1-r)^{\left(\frac{1}{m}-\frac{3}{2}\right)\alpha} \, r dr d\theta,$$

where $I < \infty$ requires

$$\left(\frac{1}{m} - \frac{3}{2}\right)\alpha + 1 > 0 \quad \Longrightarrow \quad \alpha < \frac{2m}{3m-2}.$$

Hence, for each $m \geq 1$, we can find an $\varepsilon > 0$ such that

$$\sqrt{\det(H(u))} \in L^{\frac{2m}{3m-2}-\varepsilon}(\mathrm{supp}(u)).$$

Moreover, since

$$\frac{2m}{3m-2} > \frac{2}{3},$$

we can further requires that the same ε satisfies

$$\frac{2m}{3m-2} - \varepsilon > \frac{2}{3},$$

which implies

$$\sqrt{\det(H(u))} \in L^{\frac{2m}{3m-2}-\varepsilon}(\mathrm{supp}(u)) \subset L^{\frac{2}{3}}(\mathrm{supp}(u)).$$

Hence, for each $m \geq 1$,

$$\sqrt{\det(H(u))} \in L^{\frac{2}{3}}(\mathrm{supp}(u)).$$

□

2.4 State of the art of numerical methods for solving PME

The following is a historical highlight of some significant contributions to the numerical solution of PME.

1. **Rose (1983)**

   Rose [40] analyzes a fully discrete scheme applied to PME (in dimensions 1, 2, and 3), where a linear finite element method in space and a backward Euler discretization in time are used. The analysis relies on regularizing the original PME, by perturbing the diffusion

coefficient with a parameter $\varepsilon = \mathcal{O}(h^{\frac{2m+4}{m^2+4m+2}})$, so that the resulting equation is not degenerate at the boundary of the solution's support. An error estimate is given, which says

$$\left(\sum_n \Delta t \|u_h^n - u\|^{m+2}_{L^{m+2}(\Omega)}\right)^{\frac{1}{m+2}} \leq C\left(\Delta t^{\frac{1}{m+1}} + \left(\ln\left(\frac{1}{h}\right)\right)^{\frac{1}{(m+1)(m+2)}} h^{\frac{2}{m+1}}\right), \tag{2.17}$$

where h is the maximum element diameter and u_h^n is the numerical approximation of u at $t = t_n$. The rate of convergence given is somewhat pessimistic, which indeed vanishes as m goes to infinity.

2. **Nochetto and Verdi (1988)**

In [38], the Nochetto and Verdi deal with a more general equation in the form

$$u_t - \nabla_x \cdot [\nabla_x v + b(r(v))] + f(r(v)) = 0, \tag{2.18}$$

which is not only applicable to the PME, but also the two-phase Stephan and the nonstationary filtration problems. The numerical solution is based on a discretization with linear finite element method in space, and a semi-implicit scheme in time, where the role of numerical integration is considered in the analysis. The error estimate of the method for the porous medium equation (Corollary 3 of [38, pg. 807]) is given as

$$\begin{aligned} &\|u_h - u\|_{L^\infty(0,T;H^{-1}(\Omega))} + \|(u_h)^{m+1} - u^{m+1}\|_{L^2(0,T;L^2(\Omega))} + \left\|\int_0^t ((u_h)^{m+1} - u^{m+1})\right\|_{L^\infty(0,T;H^1(\Omega))} \\ &\qquad \leq C\left(\frac{h^2}{\varepsilon} + \frac{h^4}{\varepsilon^2 \Delta t} + \Delta t\right)^{\frac{1}{2}} \\ &\qquad = \mathcal{O}(h^{\frac{m+2}{2(m+1)}}), \quad \text{if} \quad \Delta t = \mathcal{O}(h^{\frac{m+2}{m+1}}), \quad \varepsilon = \mathcal{O}(h^{\frac{m}{m+1}}), \end{aligned} \tag{2.19}$$

which is slightly better than that in [40].

3. **Socolovsky (1984)**

Socolovsky's PhD dissertation [44] presents some numerical methods for degenerate parabolic

problems having the form

$$\begin{cases} u_t - \Delta f(u) + g(u) - au = 0\,, & \text{for} \quad \boldsymbol{x} \in \Omega, \quad t > 0 \\ u(\boldsymbol{x}, t) = 0\,, & \text{for} \quad \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x})\,, & \text{for} \quad \boldsymbol{x} \in \Omega \end{cases} \tag{2.20}$$

Applications of the methods are focused on PME as a prime example. The first part of the work gives a convergence result for a finite difference scheme on (2.20) based on semigroup theory. The second part deals with the Cauchy problem of the PME in one dimension, having the form

$$\begin{cases} u_t = (|u|^m)_{xx}\,, & \text{for} \quad x \in \mathbb{R}, \quad t > 0, \quad m \geq 2 \\ u(x, 0) = u_0(x)\,, & \text{for} \quad x \in \mathbb{R}\,, \end{cases} \tag{2.21}$$

and explores numerical solutions based on finite difference and finite element methods. All the simulations are done in 1D, and only for $m = 2$ (which corresponds to $m = 1$ in the form of PME (2.8)). A development of an implicit-explicit scheme with time step $\Delta t = \mathscr{O}(h^2)$ gives optimal (second-order) convergence rate for not only the solution, but also the boundary.

4. **Ebmeyer and Liu (1998)**

Ebmeyer and Liu [16] prove a convergence result of the finite element method as applied to

$$u_t = \Delta \beta(u)\,, \tag{2.22}$$

where $\beta(\cdot)$ has to satisfy a set of rather strict requirements. This result nevertheless covers the following form of PME

$$u_t = \Delta(u|u|^{m-1})\,, \quad m > 1\,. \tag{2.23}$$

The method additionally requires that each element of the mesh be nonobtuse, i.e. the right angle is the widest that each element can have. The convergence result is an improvement over the work of [40], since the rate does not vanish as m goes to infinity. Nevertheless, the

convergence-rate in L^2-norm is not optimal even for the case $m = 2$ (corresponding to $m = 1$ in (2.8)).

5. **Baines et al. (2005)**

   An adaptive moving mesh method is developed by Baines et al. [5], which in effect conserves some local properties (such as mass) of the solution on each element of the mesh. The method is developed for a general nonlinear diffusion equation with moving boundaries in one and two dimensions. It is tested on the porous medium equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (|u|^m \nabla u), \tag{2.24}$$

   where $u = 0$ on the moving boundary and the initial solution is chosen such that the exact solution is a Barenblatt-Pattle solution given in (2.14). For the case $m = 1$, where the solution does not have a steep front, the results are optimal in both one and two dimensions. For the case $m = 3$, where the slope of the solution at the boundary is infinity, a second-order convergence is achieved for the one-dimensional case provided that the initial mesh is optimized through a complicated algorithm which effectively locates more points at the regions where the initial solution is steep. However, the same case in 2D has not been completely settled as the authors have not attempted to generate the corresponding optimal initial mesh, due to the cost of the mentioned algorithm for higher dimensions. Nevertheless, using an initial mesh with points manually clustered to the boundary, the authors achieve a convergence order greater than one, but still less than the optimal (second) order.

6. **Zhang and Wu (2009)**

   In their work, Zhang and Wu [49] apply the local discontinuous Galerkin (LDG) finite element method for PME. They prove that when the initial solution is nonnegative, the scheme preserves the non-negativity of the average of the solution on each element of the mesh provided that a parameter within the numerical scheme is chosen properly. The method, when applied to PME where the parameter m is relatively high, converges with high order in region

away from the boundary and successfully eliminates unwanted, nonphysical oscillations near the solution's free boundary. Not only so, it also pleasantly and accurately simulates some peculiar phenomena of PME, including the existence of a waiting time for a special class of initial solutions. Currently, their method is only demonstrated for the 1D setting.

7. **Duque et al. (2014)**

   Duque, Almeida, and Antontsev [14] develop a moving mesh FE method to solve a more generalized form of PME

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (u^{\gamma(\boldsymbol{x},t)} \nabla u) - u^{\sigma(\boldsymbol{x},t)}, & \text{in} \quad \Omega_T = \Omega \times (0,T] \\ u = 0, & \text{on} \quad \Gamma_T = \partial\Omega \times [0,T] \\ u(\boldsymbol{x},0) = u_0(\boldsymbol{x}), & \text{in} \quad \Omega, \end{cases} \tag{2.25}$$

   where $\partial\Omega$ is Lipschitz-continuous, and γ and σ are bounded functions on Ω_T such that

$$0 \le \gamma(\boldsymbol{x},t) \le \gamma^+ < \infty, \quad 1 \le \sigma(\boldsymbol{x},t) \le \sigma^+ < \infty, \quad \forall \boldsymbol{x} \in \overline{\Omega}_T .$$

   According to the work of Antontsev and Shmarev [1], problem (2.25) has a unique weak solution, having also the finite propagation property. Hence, a free boundary exists whenever the initial solution is compactly supported, and is moved according to a hypothesized equation

$$\Gamma'(t) = \lim_{\boldsymbol{x} \to \Gamma(t)-} -u^{(\gamma-1)} \nabla u,$$

   which indeed is very close to the standard Darcy's law. The MMPDE method (cf. [25]) for mesh generation and adaptation is utilized, where a monitor function has been chosen that depends on the gradient of the solution, which effectually moves the mesh points to regions where the slope of the solution is steep. The discretized equations for the weak formulation, boundary movement, and mesh generation are solved simultaneously. The method is applied to a specific problem of (2.25) where the exact solution is known; however, numerical results

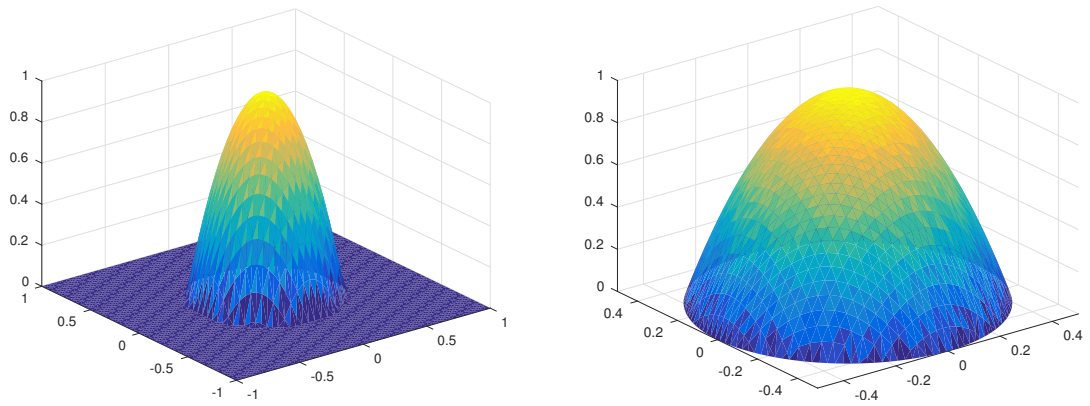
only show a first-order convergence.

2.5 The moving mesh finite element method

In this section, we would like to describe briefly our moving mesh finite element method for solving the PME. Indeed, in dealing with PDEs with free and/or moving boundaries, with PME as our special problem, there are two major approaches.

The first approach, called the "embedding" or "immersed-boundary" approach, discretizes and solves the PDE on a fixed domain which is sufficiently large to contain the support of the solution throughout the simulation (See Fig. 2.5a). The main advantage of this approach is that we are not required to trace explicitly the movement of the free boundary with Darcy's law. This also gives us the liberty of simulating PME with more complicated supports (including those that are nonconvex and those which have sharp corners and cusps). However, there is also a major disadvantage to this approach. As we have known, the finite propagation property of PME creates a free boundary which separates the nonzero and zero regions of the solution, and therefore there is oftentimes a sharp corner in the solution at the free boundary. This implies that the extended solution on this larger domain will typically be at most H^1 in its regularity, which further suggests that for a linear FE method, a first order convergence is the best that we can expect.

For the second approach, the domain of the numerical solution is dynamic, where the nodal points of the mesh are moved at each time step in order to make the overall mesh region be as close to the support of the solution as possible (See Fig. 2.5b). We can call this the nonembedding or "nonimmersed-boundary" approach. There are two great advantages with this approach. First, since we are only simulating the solution within its support where the solution's regularity is much higher, we can expect a greater order of convergence, which is up to second order for the linear FE method. The second advantage is less memory and CPU time usage, as we do not need extra mesh points to represent the zero region of the solution. One major disadvantage of the method is that at each time step, we need to use Darcy's law to move the mesh points at the free boundary outwardly,



(a) Embedding approach.

(b) Nonembedding approach.

Figure 2.5: The two typical discretization approaches for free boundary problem.

in order to simulate for the ever-expanding of the solution's support. Hence, the accuracy of the solution will be heavily affected by how accurately Darcy's law is applied. Another disadvantage of this approach is its lack of robustness in simulating PME with complicated support (mostly due to the likelihood of mesh singularities). In the following subsections, we will consider a discretization based on the first approach. The second approach is similar and will be mentioned in Subsection 2.5.4.

2.5.1 Linear Finite Element Discretization

In this section, we present a linear finite element discretization on a moving mesh for problem (2.8), where the domain Ω is chosen sufficiently large to accommodate the support of the solution throughout all time instants. Denote Ω_h as the approximation of this domain, where $\Omega_h = \Omega$ if Ω is polygonal. We would like to consider the time discretization

$$t_0 < t_1 < \ldots < t_{n_f} \equiv T. \tag{2.26}$$

Presently, we assume that at each time level $t_n, n = 0, \ldots, n_f$, Ω_h is discretized with a simplicial mesh $\mathscr{T}_h^n$, and each has the same number of nodes and elements, and with the same connectivity (the detail on how these are generated will be given in the subsection that follows). For each mesh

$\mathscr{T}_h^n$, we assume that it has N elements and N_v vertices, and denote the vertices by $\boldsymbol{x}_j^n$, $j = 1,\ldots,N_v$. Between the times t_n and t_{n+1}, the movement of the mesh is considered continuous, which allows us to represent the mesh $\mathscr{T}_h(t)$ at time $t \in (t_n, t_{n+1})$ through linear interpolation, where

$$\boldsymbol{x}_j(t) = \frac{t - t_n}{t_{n+1} - t_n}\boldsymbol{x}_j^{n+1} + \frac{t_{n+1} - t}{t_{n+1} - t_n}\boldsymbol{x}_j^n, \tag{2.27}$$

$$\dot{\boldsymbol{x}}_j(t) = \frac{\boldsymbol{x}_j^{n+1} - \boldsymbol{x}_j^n}{t_{n+1} - t_n}, \quad j = 1, ..., N_v. \tag{2.28}$$

For each time $t \in (t_0, T]$, we can define

$$\phi_j(\boldsymbol{x}, t): \quad \phi_j|_K \in \mathbb{P}_1 \quad \forall K \in \mathscr{T}_h(t), \quad \text{and} \quad \phi_j(\boldsymbol{x}_i, t) = \delta_{ij} \quad i = 1, \ldots, N_v$$

as the linear basis function associated with the vertex $\boldsymbol{x}_j(t)$. For convenience, we assume that the vertices are indexed such that the first N_{vi} vertices are the interior vertices. We denote

$$V_h(t) = \text{span}\{\phi_1(\cdot, t), ..., \phi_{N_{vi}}(\cdot, t)\}, \quad t \in (t_0, T),$$

as the finite-dimensional space of the FE solution at time $t \in (t_0, T)$. Then, the linear finite element approximation to the solution of IBVP (2.8) is defined as $u_h(\cdot, t) \in V_h(t)$, $t \in (t_0, T]$ such that

$$\begin{cases} \int_{\Omega_h} \frac{\partial u_h}{\partial t} v \, d\boldsymbol{x} = -\int_{\Omega_h} |u_h|^m \nabla u_h \cdot \nabla v \, d\boldsymbol{x}, \quad \forall v \in V_h(t), \quad t_0 < t \leq T \\ \int_{\Omega_h} (u_h(\boldsymbol{x}, 0) - u^0(\boldsymbol{x})) v \, d\boldsymbol{x} = 0, \quad \forall v \in V_h(t_0). \end{cases} \tag{2.29}$$

Since $u_h(\boldsymbol{x}, t) \in V_h(t)$, it can be expressed as a linear combination of the basis functions $\{\phi_j(\boldsymbol{x}, t)\}_{j=1}^{N_{vi}}$. Indeed, for each time $t \in (t_0, T]$, if we denote $u_j(t)$ as the approximate solution at vertex $\boldsymbol{x}_j$, then we can write

$$u_h(\boldsymbol{x}, t) = \sum_{j=1}^{N_{v_i}} u_j(t)\phi_j(\boldsymbol{x}, t). \tag{2.30}$$

Moreover, by differentiating (2.30) with respect to t, we have

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{N_{v_i}} \frac{du_j}{dt}\phi_j(\boldsymbol{x},t) + \sum_{j=1}^{N_{v_i}} u_j(t)\frac{\partial \phi_j}{\partial t}\,. \tag{2.31}$$

According to the work of Jimack and Wathen [28, Lemma 2.3], if we define the mesh velocity $\dot{\boldsymbol{X}}(\boldsymbol{x},t)$ as

$$\dot{\boldsymbol{X}}(\boldsymbol{x},t) = \sum_{j=1}^{N_v} \dot{\boldsymbol{x}}_j(t)\phi_j(\boldsymbol{x},t)\,,$$

we can show that

$$\frac{\partial \phi_j}{\partial t} = -\nabla\phi_j \cdot \dot{\boldsymbol{X}}, \quad a.e.\ \text{in}\ \Omega_h\,.$$

By this, we can simplify (2.31) into

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{N_{v_i}} \frac{du_j}{dt}\phi_j(\boldsymbol{x},t) - \nabla u_h \cdot \dot{\boldsymbol{X}}. \tag{2.32}$$

Now, we can rewrite (2.29) into a matrix form. Indeed, by substituting expressions for u_h and $\frac{\partial u_h}{\partial t}$ (from (2.30) and (2.32), respectively) into (2.29), and also by taking $v = \phi_i$ $(i = 1,...,N_{vi})$ successively, we get

$$\sum_{j=1}^{N_{vi}} \left(\int_{\Omega_h} \phi_j\phi_i \, d\boldsymbol{x} \right) \frac{du_j}{dt} = \int_{\Omega_h} \nabla u_h \cdot \left(\dot{\boldsymbol{X}}\phi_i - |u_h|^m \nabla\phi_i \right) \, d\boldsymbol{x}, \quad i = 1,...,N_{vi}, \quad t_0 < t \leq T\,, \tag{2.33}$$

which then can be cast into the matrix form of the ODE system

$$B(\boldsymbol{X})\dot{\boldsymbol{U}} = F(\boldsymbol{U},\boldsymbol{X},\dot{\boldsymbol{X}}), \tag{2.34}$$

where B is the mass matrix, $\boldsymbol{X}$ is a vector representing the mesh, and $\boldsymbol{U}$ is a vector representing the solution. Note that (2.34) is also called a system of differential algebraic equation (DAE) due to the presence and position of the mass matrix $B(\boldsymbol{X})$. In order to obtain the solution at the next time level $t = t_{n+1}$, we have chosen the fifth-order Radau IIA method [20] to solve numerically this ODE

system on time interval $[t_n, t_{n+1}]$. The step size in this Runge-Kutta method is chosen adaptively, based on a two-step error estimator of Gonzalez-Pinto et al. [19]. For these computations, we have chosen the relative and absolute tolerances as $\texttt{rtol} = 10^{-6}$ and $\texttt{atol} = 10^{-8}$, respectively.

2.5.2 The MMPDE method

In this subsection, we would like to present a method for generating adaptively a new mesh based on a current mesh and its associated solution. More specifically, if we are given the mesh $\mathscr{T}_h^n$ along with the solution u_h^n at time $t = t_n$, we seek to use the MMPDE method to generate the mesh $\mathscr{T}_h^{n+1}$ based on the given information, in a way such that the solution u_h^{n+1} at the next time level $t = t_{n+1}$ on this new mesh will be as optimal as possible.

The MMPDE (Moving mesh PDE) method, developed by Huang et al. [25], is an adaptive moving mesh method typically used in context of solving PDEs, for both time dependent and steady state problems. In essence, it is a mesh generation method based on minimizing an energy functional, so that the outcome (ideally) is an $\mathbb{M}$-uniform mesh, i.e. uniform in the Riemannian metric

$$\|\boldsymbol{x}\|_{\mathbb{M}} = \boldsymbol{x}^T \mathbb{M} \boldsymbol{x}, \quad \forall \boldsymbol{x} \in \mathbb{R}^2 \tag{2.35}$$

where the metric tensor $\mathbb{M}$ is a $d \times d$-matrix assumed to be symmetric and uniformly positive definite on Ω_h. Typically, $\mathbb{M}$ is used for controlling the size, shape, and orientation of the elements of the new mesh.

If we would like to have a mesh uniform in the natural Euclidean metric, we can choose

$$\mathbb{M} = \mathbb{I}, \tag{2.36}$$

where $\mathbb{I}$ is the $d \times d$ identity matrix. On the other hand, we can choose $\mathbb{M}$ to make our new mesh adaptive to the current solution. Indeed, there are two major adaptivity strategies pertaining to moving mesh methods. The first strategy, roughly speaking, seeks to distribute mesh points uniformly along the arclength of the solution. This strategy, which can be termed arclength-based

adaptivity, is associated with the choice of

$$\mathbb{M} = (\mathbb{I} + \nabla u_h^n (\nabla u_h^n)^T)^{\frac{1}{2}}, \tag{2.37}$$

which is derived from an error estimate of a piecewise constant interpolation. The second strategy is called Hessian-based adaptivity, where the metric tensor is given by

$$\mathbb{M} = [\det(\mathbb{I} + |H(u_h^n)|)]^{-\frac{1}{6}} (\mathbb{I} + |H(u_h^n)|), \tag{2.38}$$

where $H(u_h^n)$ is a recovered Hessian for the piecewise linear finite element solution u_h^n, and

$$|H(u_h^n)| = Q \begin{bmatrix} |\lambda_1| & & \\ & \ddots & \\ & & |\lambda_d| \end{bmatrix} Q^T, \quad \text{with} \quad H(u_h^n) = Q \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix} Q^T$$

being the eigen-decomposition of $H(u_h^n)$. This choice of $\mathbb{M}$ is optimal for minimizing the L^2 norm of the linear interpolation error [27], which results in an adaptivity that, loosely speaking, distributes the mesh points uniformly according to the solution's curvature. Here the recovered Hessian $H(u_h^n)$ is obtained through a least squares fitting strategy (e.g. [30, 31]). Additionally, if the method only discretizes on the solution's support (for the nonembedding approach), we can also consider the metric tensor

$$\mathbb{M} = \frac{1}{[(u_h^n)^2 + \alpha]^r} \mathbb{I} \tag{2.39}$$

where α and r are positive parameters, typically chosen to be 10^{-6} and $\frac{1}{2}$, respectively. This choice of $\mathbb{M}$ effectively allocates more mesh points toward the free boundary of PME solution, especially in situations where the solution is smooth (For more information, see discussion of the nonembedding methods in Chapters 4 and 5). For this reason, we term the mesh adaptivity associated with (2.39) "boundary-based."

In practice, the MMPDE method seeks to generate the new mesh $\mathscr{T}_h^{n+1}$ to be as $\mathbb{M}$-uniform as

possible. Roughly speaking, we would like for all the elements of $\mathscr{T}_h^{n+1}$, under this special metric, to be proportional-in-size and similar-in-shape to their corresponding reference elements in a reference mesh $\hat{\mathscr{T}}_{c,h}$, typically chosen as the initial physical mesh $\mathscr{T}_h(t_0) = \mathscr{T}_h^0$. These fundamental ideas can be quantified precisely into the equidistribution and alignment conditions [23, 26]

$$|K|\sqrt{\det(\mathbb{M}_K)} = \frac{\sigma_h |K_c|}{|\Omega_c|}, \quad \forall K \in \mathscr{T}_h \tag{2.40}$$

$$\frac{1}{d}\text{trace}\left(\left(F_K'\right)^{-1}\mathbb{M}_K^{-1}\left(F_K'\right)^{-T}\right) = \det\left(\left(F_K'\right)^{-1}\mathbb{M}_K^{-1}\left(F_K'\right)^{-T}\right)^{\frac{1}{d}}, \quad \forall K \in \mathscr{T}_h \tag{2.41}$$

where $|K|$ and $|K_c|$ are the volumes of K and its corresponding reference element $K_c \in \mathscr{T}_{c,h}$, respectively, $\det(\cdot)$ and $\text{trace}(\cdot)$ denote the determinant and trace of a matrix, respectively, $\mathbb{M}_K$ is the average of $\mathbb{M}$ over K (i.e. $\mathbb{M}_K = \frac{1}{|K|}\int_K \mathbb{M}(\boldsymbol{x})\, d\boldsymbol{x}$), $F_K' = \partial F_K/\partial \boldsymbol{x}$ is the Jacobian matrix of the affine mapping $F_K : K_c \to K$, and

$$\sigma_h = \sum_{K\in\mathscr{T}_h^n} |K|\sqrt{\det(\mathbb{M}_K)}, \quad |\Omega_c| = \sum_{K_c\in\hat{\mathscr{T}}_{c,h}} |K_c|.$$

In fact, according to [22], we can obtain our new mesh $\mathscr{T}_h^{n+1}$ by minimizing the energy function

$$\begin{aligned} I_h = \theta \sum_{K\in\mathscr{T}_h^n} |K|\sqrt{\det(\mathbb{M}_K)}\left(\text{trace}((F_K')^{-1}\mathbb{M}_K^{-1}(F_K')^{-T})\right)^{\frac{dp}{2}} \\ + (1-2\theta)d^{\frac{dp}{2}} \sum_{K\in\mathscr{T}_h^n} |K|\sqrt{\det(\mathbb{M}_K)}\left(\frac{|K_c|}{|K|\sqrt{\det(\mathbb{M}_K)}}\right)^p, \end{aligned} \tag{2.42}$$

where $\theta \in (0, \frac{1}{2}]$ and $p > 1$ are non-dimensional parameters. Equation (2.42) is the discrete form of the functional developed in [22], where the equidistribution and alignment conditions are combined in variational mesh adaptation. For our computations, we have chosen $\theta = 1/3$ and $p = 2$, which tend to produce better results, based on experience.

We notice that I_h is a function of only $\boldsymbol{\xi}_j$, $j = 1, ..., N_v$, and also observe that direct minimization

problem

$$\operatorname{args}\min_{\{\boldsymbol{\xi}_j\}_{j=1}^{N_v}} I_h[\boldsymbol{\xi}_j]$$

is highly nonlinear, which is very difficult, if not impossible, to solve directly. Hence, instead of solving the minimization directly, we treat I_h as the gradient flow with respect to variables $\{\boldsymbol{\xi}_j\}_{j=1}^{N_v}$, which takes the form of the MMPDE [25]

$$\frac{d\boldsymbol{\xi}_j}{dt} = -\frac{P_j}{\tau}\left[\frac{\partial I_h}{\partial \boldsymbol{\xi}_j}\right]^T, \quad j = 1,...,N_v \tag{2.43}$$

where the row vector $\partial I_h/\partial \boldsymbol{\xi}_j$ is the derivative of I_h with respect to $\boldsymbol{\xi}_j$, $\tau > 0$ is a parameter used to dictate how fast the mesh movement will react to any change in the metric tensor, and $P_j = \det(\mathbb{M}(\boldsymbol{x}_j))^{\frac{p-1}{2}}$ is chosen such that (2.43) is invariant under the scaling transformation of $\mathbb{M}$: $\mathbb{M} \to c\mathbb{M}$ for any positive constant c. The recent work of Huang and Kamenski [24] presents a method to calculate the derivative of I_h with respect to $\boldsymbol{\xi}_j$ through the notion of scalar-by-matrix differentiation, where the authors have developed important analytical formulas in dealing with our specific situation. With such analytical formulas, we can rewrite (2.43) into

$$\frac{d\boldsymbol{\xi}_j}{dt} = \frac{P_j}{\tau}\sum_{K\in\omega_j} |K|\boldsymbol{v}_{j_K}^K, \quad j = 1,\ldots,N_v \tag{2.44}$$

where ω_j is the element patch associated with the vertex $\boldsymbol{x}_j$, j_K denotes the local index of the same vertex on $K \in \omega_j$, and $\boldsymbol{v}_{j_K}^K$ is the velocity contributed by K to the vertex with local index j_K of the same element. The velocities contributed by K to its vertices are given by

$$\begin{bmatrix} (\boldsymbol{v}_1^K)^T \\ \vdots \\ (\boldsymbol{v}_d^K)^T \end{bmatrix} = -E_K^{-1}\frac{\partial G}{\partial \mathbb{J}} - \frac{\partial G}{\partial \det(\mathbb{J})}\frac{\det(\hat{E}_K)}{\det(E_K)}\hat{E}_K^{-1}, \quad \boldsymbol{v}_0^K = -\sum_{i=1}^{d}\boldsymbol{v}_i^K, \tag{2.45}$$

where $E_K = [\boldsymbol{x}_1^K - \boldsymbol{x}_0^K,...,\boldsymbol{x}_d^K - \boldsymbol{x}_0^K]$ and $\hat{E}_K = [\boldsymbol{\xi}_1^K - \boldsymbol{\xi}_0^K,...,\boldsymbol{\xi}_d^K - \boldsymbol{\xi}_0^K]$ are the edge matrices of K and

K_c, respectively, $\mathbb{J} = (F_K')^{-1}$, the function G is associated with the energy (2.42) and defined as

$$G(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}) = \theta \sqrt{\det(\mathbb{M})} \left(\operatorname{trace}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}} + (1-2\theta) d^{\frac{dp}{2}} \sqrt{\det(\mathbb{M})} \left(\frac{\det(\mathbb{J})}{\sqrt{\det(\mathbb{M})}} \right)^p,$$

and its derivatives (evaluated at $(\mathbb{J}, \det(\mathbb{J}), \mathbb{M}) = ((F_K')^{-1}, \det(F_K')^{-1}, \mathbb{M}_K)$) with respect to the first and second arguments are given by

$$\begin{aligned}
\frac{\partial G}{\partial \mathbb{J}} &= dp\theta \sqrt{\det(\mathbb{M})} \left(\operatorname{trace}(\mathbb{J}\mathbb{M}^{-1}\mathbb{J}^T)\right)^{\frac{dp}{2}-1} \mathbb{M}^{-1}\mathbb{J}^T, \\
\frac{\partial G}{\partial \det(\mathbb{J})} &= p(1-2\theta) d^{\frac{dp}{2}} \det(\mathbb{M})^{\frac{1-p}{2}} \det(\mathbb{J})^{p-1}.
\end{aligned}$$

We notice that $\partial G / \partial \mathbb{J}$ is a d-by-d matrix.

For the mesh points lying on the boundary, the mesh equation (2.44) should be modified accordingly. Typically, for fixed points on boundary, the velocities can be set to zero. If the boundary points are allowed to move (or slide), the velocities associated with these should be modified so that they do not move outside the domain. With these formulas and settings, the mesh equation (2.44), with $\hat{\mathscr{T}}_{c,h}$ as the initial mesh, can be integrated from $t = t_n$ to $t = t_{n+1}$ by an ODE solver in order to obtain the new computational mesh $\mathscr{T}_{c,h}^{n+1}$. Here, we use the Numerical Differentiation Formula (NDF) based implicit scheme `ode15s` of Matlab for the purpose. Since we have assumed nonsingularity in the new mesh, and that it also has the same number of elements and vertices, and the same connectivity as the physical mesh $\mathscr{T}_h^n$ at time $t = t_n$, there exists a 1-to-1 correspondence $\mathscr{T}_h^n = \Phi_h^n(\mathscr{T}_{c,h}^{n+1})$. The new physical mesh at time $t = t_{n+1}$ is then defined and computed (via linear interpolation) as $\mathscr{T}_h^{n+1} = \Phi_h^n(\hat{\mathscr{T}}_{c,h})$.

2.5.3 Solution procedure for the embedding approach

From the discussion of Subsections 2.5.1 and 2.5.2, we may summarize our method for PME with the embedding approach as following:

1. Establish a domain Ω_h sufficiently large to cover the solution's support at the final time,

and discretize it with a physical mesh $\mathscr{T}_h^0$. Here, we let $\hat{\mathscr{T}}_{c,h} := \mathscr{T}_h^0$ be our computational reference mesh for the MMPDE method.

2. Assume that at time level $t = t_n$, we have the solution u_h^n on the mesh $\mathscr{T}_h^n$. By applying the MMPDE method, we can generate a new mesh $\mathscr{T}_h^{n+1}$ for the next time level $t = t_{n+1}$.

3. By the moving mesh FE method (detailed in Subsection 2.5.1), we can compute the solution u_h^{n+1} based on u_h^n, $\mathscr{T}_h^n$, and $\mathscr{T}_h^{n+1}$.

4. Repeat the procedure in steps 2 and 3 until the final time $t = t_{n_f}$ is reached.

Notice that, since the set of mesh equations (2.44) has a different structure than that of the physical equations (2.34), we deem it prudent to solve them separately.

2.5.4 Discretization and mesh movement for the nonembedding method

The previous two Subsections 2.5.1 and 2.5.2 present the FE discretization on a moving mesh along with the MMPDE method for mesh generation. In particular, they deal with problem (2.8) where the domain Ω_h is assumed to be fixed and also sufficiently large to accommodate the support of the solution throughout the simulation. On the other hand, the nonembedding approach (as seen, for example, in the work of [5, 14]), discretizes only on the support of the solution at each time; it can also be called the "nonimmersed-boundary" approach (cf. Pg. 29). In order to apply this approach, we need to consider a modified version of the original IBVP (2.8)

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u)\,, & \text{in} \quad \Omega(t)\,, \quad t \in (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x})\,, & \text{in} \quad \Omega(t_0) \\ u(\boldsymbol{x}, t) = 0\,, & \text{on} \quad \Gamma(t) := \partial\Omega(t)\,, \quad t \in (t_0, T] \\ \Gamma'(t) = \lim\limits_{\boldsymbol{x} \to \Gamma(t)^-} -\nabla \left(\dfrac{|u|^m(\boldsymbol{x}, t)}{m} \right) \cdot \hat{\boldsymbol{n}}\,, & \text{in} \quad (t_0, T]\,, \end{cases} \tag{2.46}$$

where the domain $\Omega(t)$ depends on time and is the same as $\text{supp}(u(\boldsymbol{x}, t))$, and where Darcy's law is incorporated to account for movements of the free boundary. The discretization procedure for

(2.46) is very similar to the embedding method in Subsection 2.5.1. However, there are expected differences. Suppose that at time level $t = t_n$, we are given the mesh $\mathscr{T}_h^n$ (which represents the domain $\Omega_h^n \approx \Omega(t_n)$) and its associated solution u_h^n. The generation of the new mesh $\mathscr{T}_h^{n+1}$ at time level $t = t_{n+1}$, based on $\mathscr{T}_h^n$ and u_h^n, is done similarly as in subsection 2.5.2. However, before the MMPDE method can be applied, we first need to move the boundary points of $\mathscr{T}_h^n$ manually via Darcy's law. This boundary velocity formula, as seen in (2.46), can be discretized with the forward Euler scheme as

$$\frac{\Gamma^{n+1} - \Gamma^n}{t_{n+1} - t_n} = -\nabla_h \left(\frac{(u_h^n)^m}{m} \right) \cdot \hat{\boldsymbol{n}}, \tag{2.47}$$

where $\nabla_h(\cdot)$ represents a scheme for approximating $\lim_{\boldsymbol{x} \to \Gamma(t_n)^-} \nabla(\cdot)$ at time $t = t_n$, and $\hat{\boldsymbol{n}}$ denotes the unit outward normal to the boundary (see an illustration of the outward normal directions to a domain in Fig. 2.6). The MMPDE method then can be applied, from which we have the new mesh $\mathscr{T}_h^{n+1}$ at time level $t = t_{n+1}$.

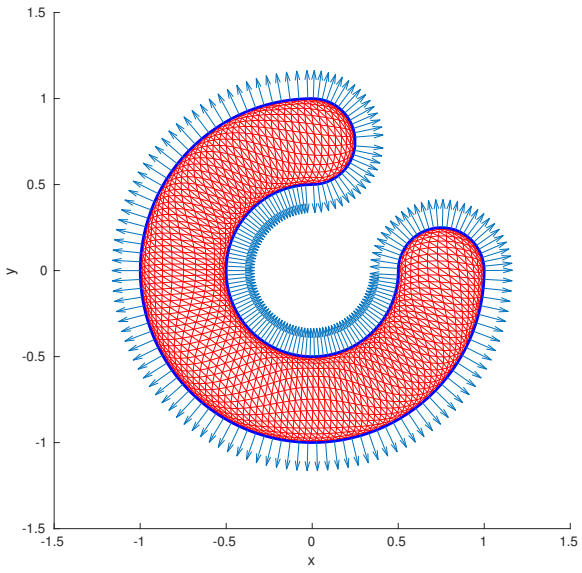


Figure 2.6: The outward normal directions to each boundary point of a PME solution.

In order to find the FE solution u_h^{n+1} at the next time level, the integrals involved in the weak formulation as seen in subsection 2.5.1 will be performed on the time-dependent domain $\Omega_h^{n+1} \approx$

$\Omega(t_{n+1})$. For example, the system of equations in (2.33) should be modified into

$$\sum_{j=1}^{N_{vi}} \left(\int_{\Omega_h^{n+1}} \phi_j \phi_i \, d\boldsymbol{x} \right) \frac{du_j}{dt} = \int_{\Omega_h^{n+1}} \nabla u_h \cdot \left(\dot{\boldsymbol{X}} \phi_i - |u_h|^m \nabla \phi_i \right) \, d\boldsymbol{x}, \quad i = 1, ..., N_{vi}, \quad t_0 < t \leq T \, .$$

In summary, we need to solve the follow set of equations

$$\begin{cases} \boxed{\text{Boundary Equations}} \\ \boxed{\text{Mesh Equations}} \\ \boxed{\text{Physical Equations}} \end{cases} \tag{2.48}$$

However, since the mesh and the solution systems in general have different structures, and their coupling is highly nonlinear, we would like to split the solution strategy via the *BMP-procedure* (i.e. "Boundary-Mesh-Physical") as following

1. Discretize $\Omega(t_0)$ into Ω_h^0 with the mesh $\mathscr{T}_h^0$. Here we let $\hat{\mathscr{T}}_{c,h} := \mathscr{T}_h^0$ as our reference mesh for the MMPDE method. Assume that at time level $t = t_n$, we have the solution u_h^n on the mesh $\mathscr{T}_h^n$.

2. At time level $t = t_n$, apply the scheme for Darcy's law (2.47) to the boundary Γ^n to obtain the new boundary Γ^{n+1}. This new boundary reflect the new domain Ω_h^{n+1}, and the physical mesh changes to $\tilde{\mathscr{T}}_h^{n+1}$.

3. With the mesh $\tilde{\mathscr{T}}_h^{n+1}$ and its corresponding solution u_h^n, we can apply the MMPDE method to get the new physical mesh $\mathscr{T}_h^{n+1}$ at time level $t = t_{n+1}$.

4. Apply the moving mesh FE method (detailed in Subsection 2.5.1) with u_h^n, $\mathscr{T}_h^n$, and $\mathscr{T}_h^{n+1}$ to obtain the new solution u_h^{n+1} at time level $t = t_{n+1}$.

5. Repeat steps 2 to 4 until the final time $t = t_{n_f}$ is reached.

Chapter 3

The UE-Method: An embedding numerical solution for the U-formulation of PME

In this chapter, we present some numerical observations and results from applying the moving mesh method developed in Chapter 2 to the embedded U-formulated IBVP

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u), & \text{in} \quad \Omega \times (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x}), & \text{on} \quad \Omega \\ u(\boldsymbol{x}, t) = 0, & \text{on} \quad \partial\Omega \times (t_0, T], \end{cases} \tag{3.1}$$

where the fixed domain Ω is sufficiently large as to contain the solution's support at all time $t \in (t_0, T]$. For convenience, this approach can be termed the "UE-method" (where "U" stands for the U-formulation, and "E" the embedding approach). One of the major advantages of this embedding approach is that we do not need to explicitly trace the boundary via Darcy's law (cf. (2.13)) at each time level (This task has been a major challenge for our research, as we will see in Chapter 4.) This consequently enables us to apply the method to PME with more complex supports, including those that are nonconvex, and having tricky corners and cusps, as will be seen in the numerical examples in Sections 3.1, 3.2, and 3.3.

3.1 The Barenblatt-Pattle numerical solution

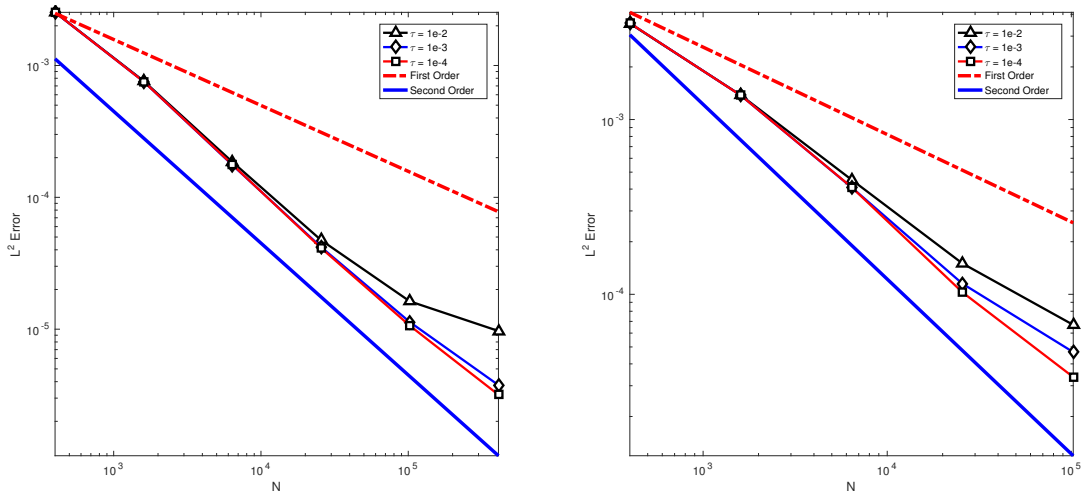
Our research findings indicate that parameter τ (cf. (2.44)), parameter m of PME, and the choice of the metric tensor $\mathbb{M}$ play sensitive roles in our numerical method, and we would like to illustrate the effects of these in this section. In the following experiments, we measure the accuracy of the UE-method using the Barenblatt-Pattle solution (2.14), where the error $|u_h - u|$ of the numerical solution is considered in the global L^2 norm, namely

$$\|u_h - u\|_{L^2(t_0,T;L^2(\Omega))} = \left(\int_{t_0}^{T} \int_{\Omega} (u_h - u)^2(\boldsymbol{x},t) d\boldsymbol{x} dt \right)^{\frac{1}{2}}. \tag{3.2}$$

We have chosen the L^2-norm since various works on numerical solution for PME have utilized the same norm in their error estimates (e.g. (1.6) of [16]). Unless otherwise noticed, we use $r_0 = 0.5$, $T = (t_0 + 0.1)/2$, $\tau = 10^{-4}$, and $\Delta t_{max} = 10^{-4}$ in the subsequent numerical experiments.

Recall that τ is a parameter which is used in the MMPDE (2.44) to adjust the responsiveness of mesh movement to any change in the metric tensor $\mathbb{M}(\boldsymbol{x},t)$ (cf. Subsection 2.5.2). In order to see the effect of this parameter on our method, we apply the numerical procedures on Hessian-based adaptive meshes (which, as we shall demonstrate later, are more optimal for accuracy and convergence order). Fig. 3.1 shows convergence histories for $\tau = 10^{-2}, 10^{-3}$, and 10^{-4}, with the cases of $m = 1$ and $m = 2$. It is noticeable that for both of these cases, when $\tau = 10^{-2}, 10^{-3}$, the convergence order decreases as the meshes become finer, whereas a second order of convergence is steadfastly held for $\tau = 10^{-4}$ over the same range of mesh sizes. This phenomenon suggests that in order to maintain a decent accuracy and convergence order, the concentration of mesh points due to Hessian-based adaptivity needs to follow very closely at a sufficient speed with the movement of the free boundary.

We also would like to know how the method is affected, when applied under each of the three meshing strategies (cf. Subsection 2.5.2). To this end, we conduct several experiments with the method where τ is chosen sufficiently small (i.e. $\tau = 10^{-4}$), and where the maximum allowed time step $\Delta t_{max} \leq 10^{-2}$. Convergence histories for the cases $m = 1$ and $m = 2$ are given in Fig. 3.2. For



(a) $m = 1$ (b) $m = 2$

Figure 3.1: Convergence history for different values of τ.

the case $m = 1$, where the solution does not have a steep slope at the free boundary, the order of convergence is approximately 1.5 (i.e. of $\mathscr{O}\left(1/\sqrt{N}\right)^{1.5}$) for the uniform and the arclength-based adaptive meshes (with little or no distinction in performance between the two). However, for the case $m = 2$, where the slope of the solution is very steep/infinite near/at the free boundary, the order of convergence is reduced to 1 for these two same meshing strategies, with the arclength-based mesh producing a slightly more accurate solution than the other. On the other hand, for the Hessian-based adaptive meshes, there is a major breakthrough as we have observed a second-order convergence for both cases, as seen in Fig. 3.2.

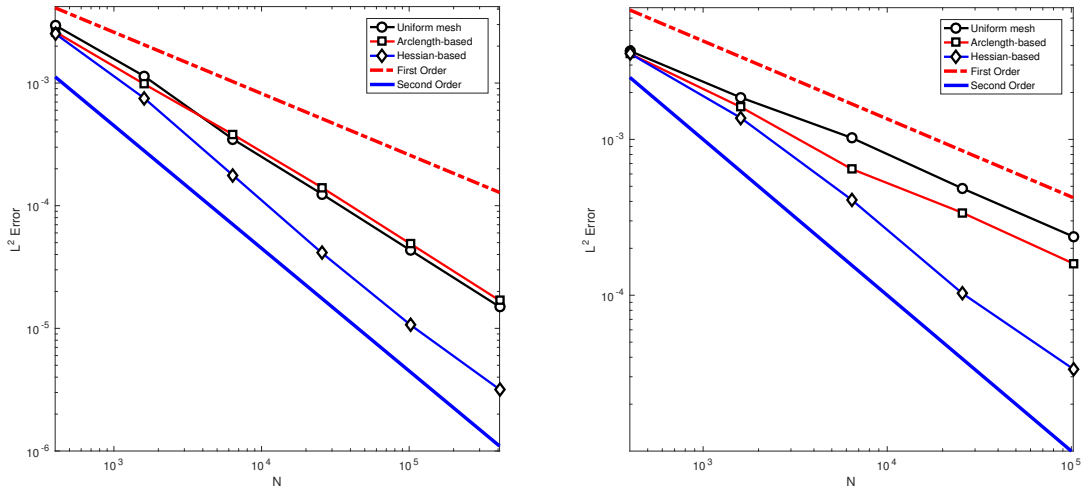
We know from Theorem 2.3.1 that a Barenblatt-Pattle solution u is only $H^1(\text{supp}(u))$ for $m = 1$, and is even less regular for $m > 1$ (i.e. $u \in W^{1,\frac{m}{m-1}-\varepsilon}(\text{supp}(u))$, for some $\varepsilon > 0$). From the standard result for FE method on a fixed mesh, we do not expect the order of convergence of our method for these special solutions to be higher than one. Yet despite all these, our method on uniform and arclength-based moving meshes have shown a higher-than-expected convergence rate (of 1.5) for $m = 1$, and more surprisingly the optimal second-order convergence for both cases of $m = 1$ and $m = 2$ for Hessian-based adaptive meshes. At the present, though we do not have rigorous justifications for this result, we would like to make two relevant observations which support it

to certain degree. First, we observe that the Hessian-based adaptivity concentrates mesh points more densely around the free boundary than the uniform and arclength-based meshing strategies; this is illustrated by the corresponding representative meshes in Fig. 3.3. Secondly, though the Barenblatt-Pattle solution has a lower regularity when considered on a larger domain containing the free boundary (cf. Theorem 2.3.1, pg. 20), it is indeed infinitely smooth within its support (i.e for each compact set $K \in \text{supp}(u)$, $u \in C^{\infty}(K)$). Moreover, from the work of [27], it is known that the error from a linear interpolation on an $\mathbb{M}$-uniform mesh (cf. Pg. 33) of a polygonal domain D, where $\mathbb{M}$ associates with the Hessian-based adaptivity (as shown in (2.38)), is given by

$$\|u - \Pi_1 u\|_{L^2(D)} \leq CN^{-1} \left\| \sqrt{\det(|H(u)|)} \right\|_{L^{\frac{2}{3}}(D)} + h.o.t., \tag{3.3}$$

where $h.o.t.$ stands for higher order terms. From Theorem 2.3.2, we know that equation (3.3) is indeed satisfied for Barenblatt-Pattle solutions *within* their positive regions (i.e. their supports). By this, we may expect a second-order convergence (i.e. of $\mathscr{O}(N^{-1})$) of the method for these solutions if the linear interpolation is only considered within their support. Even though this observation does not apply directly to our situation, where the domain is larger than the solution's support, it offers some insights into the observed phenomenon.

So far, we have seen how parameters τ and the three meshing strategies affect our solution. We would like to perform a similar evaluation for PME parameter m. As we have learned from section 2.3, that the higher m is, the steeper the gradient of a Barenblatt-Pattle solution will be at its free boundary, which will make numerical simulation more challenging. Indeed, on a uniform mesh, the order of convergence declines as m gets higher, as one can see from the convergence histories in Fig. 3.4a for cases $m = 1, 2$, and 3. This is indeed qualitatively consistent with many FE error estimates on quasi-uniform meshes found in literature on numerical solution of PME, as exemplified by Eqn. (1.3), (1.4), (1.6), and (1.7) (see also [16, 17, 38, 40]). However, for the Hessian-adaptive meshes, a second-order convergence is observed for all of these three cases as seen in Fig. 3.4b, which suggests that the rate of convergence is independent of m, even though the



(a) $m = 1$ (b) $m = 2$

Figure 3.2: Convergence history for the three meshing strategies as N (the number of the elements) increases.

magnitude of errors gets higher as m increases.

Example 3.1.1 (Barenblatt-Pattle solution). We apply the UE-method on a Hessian adaptive mesh with $N = 25600$,

$$\tau = 10^{-4}, \quad \text{and} \quad \Delta t_{max} = 10^{-3}$$

to (1.2) where

$$m = 3, \quad r_0 = 0.5, \quad \text{and} \quad T = 0.1.$$

The final mesh along with the computed solution are given in Fig. 3.5. Notice the very steep slope of the solution at the free boundary at the final time. This irregularity in the solution has caused some unwanted oscillations at the boundary, as seen in Fig. 3.6. A solution toward removing these oscillations could be an implementation of the Discontinuous Galerkin method (as in [49]) or a monotone scheme which preserves the discrete maximum principle (as in [37]).

3.2 Application to problems with complex solution's support

In their work [49], Zhang and Wu develop a numerical method for PME via the local discontinuous Galerkin (LDG) approach. Not only is their method quite accurate (up to order three within the support of solution), it is also quite flexible in simulating PME solutions with complex support and those which exhibit the waiting time phenomenon. Since the numerical results shown in [49] are all in 1D, we would like to experiment our UE-method with the 2D equivalents of some of these examples. We are motivated to do so by the fact that because the method does not require the explicit tracing of the free boundary via Darcy's law, it will leave much room and flexibility for simulating solutions with extraordinary geometry in their supports. We will demonstrate the idea in the next three examples.

Example 3.2.1. The first example models the movement and interaction of two columns of a substance (e.g. ideal gas), which are initially separate and with the same height. The model is the IBVP (1.2) with

$$m = 5, \quad \Omega = (-5.5, 5.5) \times (-5.5, 5.5),$$

and

$$u_0(x,y) = \begin{cases} 1, & \text{for} \quad (x,y) \in (0.5,3) \times (0.5,3) \\ 1, & \text{for} \quad (x,y) \in (-3,-0.5) \times (-3,-0.5) \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Here, we apply our UE-method with Hessian-based adaptivity, and a typical adaptive mesh and the corresponding numerical solution for selected time instants are shown in Fig. 3.8. We can observe that as time evolves, the support of the two columns of substance expands monotonically (i.e. it is noncontracting, cf. Pg. 17), and merges into one bigger region in a finite time. Moreover, since the slope of the solution is steep at the free boundary, the adaptive method dictates and maintains a concentration of mesh points around the free boundary throughout the time. It is important to note that the mesh remains nonsingular for the whole time, especially during the period when the two separate columns merge into one. Furthermore, we observe that the free boundary of the solution,

though starts out with some square corners, becomes smoother in the progression of time; this phenomenon has been discovered and proven in PME literature (e.g., see [42, 46]). □

Example 3.2.2. The following example is similar to the previous one, however with a slightly modified initial solution

$$u_0(x,y) = \begin{cases} 1, & \text{for} \quad (x,y) \in (0.5,3) \times (0.5,3) \\ 1.5, & \text{for} \quad (x,y) \in (-3,-0.5) \times (-3,-0.5) \\ 0, & \text{otherwise}, \end{cases} \tag{3.5}$$

so that one of the columns of substance is given some extra height. A typical adaptive mesh and the corresponding solution at selected time instants are shown in Fig. 3.9. As before, the mesh points are attracted to and remain around the free boundary as time evolves. But unlike the previous example where the two columns start with same height and expands with the same rate in their supports, in this example, the support of the column with greater initial height expands faster than the other. This indeed makes the overall support of the solution expand faster than that of the previous example; in particular, the merging of the two separate columns takes place at an earlier time than that in Example 3.2.1. □

Example 3.2.3. In this example, we consider a smooth solution on a donut-like nonconvex domain first seen in the work of Baines et al. [5]. The model is the IBVP (1.2) with

$$m = 2, \quad \Omega = (-2,2) \times (-2,2),$$

and

$$u_0(x,y) = \begin{cases} \sqrt{0.25^2 - (\sqrt{x^2+y^2} - 0.75)^2}, & \text{for} \quad \sqrt{x^2+y^2} \in [0.5,1] \text{ and } (x<0 \text{ or } y<0) \\ \sqrt{0.25^2 - x^2 - (y-0.75)^2}, & \text{for} \quad x^2 + (y-0.75)^2 \leq 0.25^2 \text{ and } x \geq 0 \\ \sqrt{0.25^2 - (x-0.75)^2 - y^2}, & \text{for} \quad (x-0.75)^2 + y^2 \leq 0.25^2 \text{ and } y \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

Here, we apply our UE-method with Hessian-based adaptivity, and typical adaptive meshes and their corresponding numerical solutions are shown in Fig. 3.10. Here, we observe the self merging/closing of the support, for which the method experiences no mesh singularity. □

Example 3.2.4 (Waiting-time phenomenon). From Subsection 2.2.2 (pg. 18), we recall that PME exhibits the waiting-time phenomenon for a certain type of initial solutions. To see this, we consider an IBVP

$$\begin{cases} u_t = \nabla \cdot (8|u|^7 \nabla u), & \text{in} \quad \Omega \times (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x}), & \text{on} \quad \Omega \\ u(\boldsymbol{x}, t) = 0, & \text{on} \quad \partial\Omega \times (t_0, T] \end{cases} \tag{3.7}$$

with

$$\Omega = (-\pi, \pi) \times (-\pi, \pi),$$

and the initial solution

$$u_0(x,y) = \begin{cases} \cos(\sqrt{x^2+y^2}), & \text{for} \quad \sqrt{x^2+y^2} \leq \frac{\pi}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

Notice that the PME in (3.7) has been slightly modified from its original form in IBVP (1.2), in order to make this example comparable to its 1D counterpart in the work of Zhang and Wu [49].

We observe that

$$\nabla \cos^7(\sqrt{x^2+y^2}) = -\frac{7\cos^6(\sqrt{x^2+y^2})\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} \begin{bmatrix} x \\ y \end{bmatrix}$$

vanishes at $\sqrt{x^2+y^2} = \frac{\pi}{2}$, and hence by Darcy's law (2.13), we do not expect the free boundary to move initially. Fig. 3.12 shows a few typical adaptive meshes and their corresponding solutions. In Fig. 3.11, we also show the cross section in the plane $y = 0$ of a computed solution at various time instants. A closer look (with the dash lines in Fig. 3.11 for indicating the initial free boundary) suggests that the free boundary of the solution does not move until around $t = 10$. Before the waiting time expires, the solution is steepening over a smaller region within the support. Such region of steepening continues to expand to eventually cover the entire support. Then, when the solution is sufficiently steep at the boundary, the support begins to expand (i.e. the free boundary begins to move), marking the end of the waiting time period. The results are consistent with the waiting time example in [49]. □

3.3 Numerical experiment for PME with variable exponent and absorption

In the previous section, we have demonstrated the robustness of our embedding moving mesh method with respect to its ability to simulate PME solutions with more complex supports. In this section, we would like to demonstrate that the method's robustness also extends to more general forms of the IBVP (1.2) as well. In particular, we will apply our method to PME with absorption and/or variable exponents,

$$u_t = \nabla \cdot (u^\gamma \nabla u) - \lambda u^\sigma, \quad \Omega \times (t_0, T] \tag{3.9}$$

subject to a homogeneous Dirichlet boundary condition and an initial condition. Here, $\gamma = \gamma(\boldsymbol{x},t)$ and $\sigma = \sigma(\boldsymbol{x},t)$ are nonnegative bounded functions and λ is a constant. PME in the form of (3.9) arises in continuum mechanics to model the motion of a barotropic gas through a porous medium, where the pressure is dependent on the density and temperature [1].

Like the standard PME in (1.2), PME (3.9) with constant exponents has been studied extensively; e.g., see [32, 42]. Nevertheless, very few theoretical results have been found for the case with variable exponents [1, 34]; for example, there is no theoretical result on the movement of the free boundary (cf. (2.13)), although the solution to (3.9) is known to have the property of finite propagation. In addition, there have been very few numerical works regarding such situation, besides the very recent works of Duque et al. [13, 14, 15].

Example 3.3.1 (Constant exponents with absorption). We first consider an example with an absorption term, in particular the PDE (3.9) with

$$\lambda = 1, \quad \gamma = 2, \quad \sigma = 0.1, \quad \Omega = (-1.5\pi, 1.5\pi) \times (-1.5\pi, 1.5\pi),$$

and the initial solution

$$u_0 = \begin{cases} |\sin(\sqrt{x^2+y^2})|, & \text{for} \quad \sqrt{x^2+y^2} \in (\frac{\pi}{6}, \pi) \\ 0.5, & \text{for} \quad \sqrt{x^2+y^2} \in [0, \frac{\pi}{6}) \\ 0, & \text{otherwise.} \end{cases}$$

This example is the two-dimensional generalization of a one-dimensional example in [49] that shows a splitting phenomenon in the middle of the solution after a finite time. Once again, the UE-method with Hessian-based adaptivity is applied, by which we can observe the splitting phenomenon as shown through the adaptive meshes and corresponding numerical solutions in Fig. 3.13. Indeed, as time evolves, we see that the solution gets lower and the support expands on the outer boundary, and the initial dimple on the top eventually goes down and "punches" through the *xy*-plane, making a visible hole in the support. This exhibits an interesting feature due to a

presence of the absorption term. □

Example 3.3.2 (Variable exponent without absorption). The following example considers a "hole-filling" phenomenon exhibited by PDE (3.9) where

$$\lambda = 0, \quad \gamma = \left(\frac{x}{2}\right)^2 + \left(\frac{y}{2}\right)^2 + 1.1, \quad \Omega = (-2,2) \times (-2,2),$$

with initial solution

$$u_0 = \begin{cases} -\sin(2\pi\sqrt{x^2+y^2}), & \text{for} \quad 0.5 < \sqrt{x^2+y^2} < 1 \\ 0, & \text{otherwise.} \end{cases}$$

This example has been studied and simulated by Duque et al. in [13, 15], where the support of the solution has a hole in the middle which disappears in a finite time. In computation, we again use our UE-method with a Hessian-based adaptivity, by which we observe the mentioned phenomenon as seen through the adaptive meshes and corresponding numerical solutions in Fig. 3.14 over the time period of $t \in [0, 0.2]$.

The result appears to have better resolution than that in [13] where a uniform mesh has been used. Moreover, our method works just fine through the closing of the inside hole (cf. Fig. 3.14) whereas the method in [15] which explicitly traces the free boundary encounters the mesh singularity problem near the time when the hole is closing. □

Example 3.3.3 (Waiting-time for variable exponent without absorption). The following example, which has been studied in [15], considers a PME with variable exponent where a waiting time phenomenon (similar to Example 3.2.4) is also exhibited. Here the model is of IBVP (1.2) for $t \in [0, 1]$, where

$$\lambda = 0, \quad \gamma = 2 - x - y, \quad \Omega = (-1.5, 1.5) \times (-1.5, 1.5),$$

with an initial solution

$$u_0 = \begin{cases} 5(0.25 - x^2 - y^2), & \text{for} \quad \sqrt{x^2+y^2} < 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

The UE-method with Hessian-based adaptivity is used, from which we have the moving meshes and corresponding numerical solutions shown in Fig. 3.15. We can see that the given variation of the exponent causes the free boundary to expand anisotropically and the solution to have different steepness along the free boundary. Moreover, a closer examination of the results confirms the waiting time phenomenon, where the interface in the region $\{(x,y) : x+y \leq 0\}$ does not move until a finite time has elapsed. Fig. 3.16 show the cross sections of the numerical solutions in the plane $y = x$ at various instants of time. In the figure, the dashed line refers to the position of the initial interface, where the waiting time phenomenon eventually occurs. □

Example 3.3.4 (Variable exponents with absorption). The following example, taken from [14], considers the IBVP (3.9) which has time dependent exponents, where

$$\lambda = 1, \quad \gamma = \frac{x^2+y^2}{t^2+1}, \quad \sigma = x^2+y^2+1+e^{-t}, \quad \Omega = (-1.5, 1.5) \times (-1.5, 1.5),$$

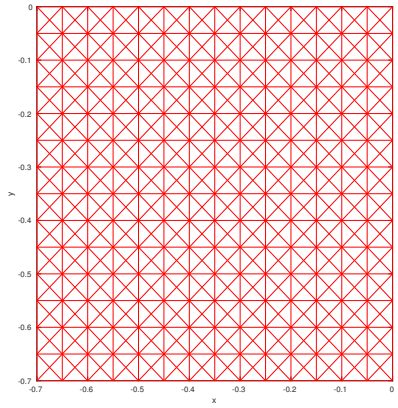
with initial solution

$$u_0 = \begin{cases} \cos(2\pi(x^2+y^2)), & \text{for} \quad \sqrt{x^2+y^2} < 0.5 \\ 0, & \text{otherwise.} \end{cases}$$

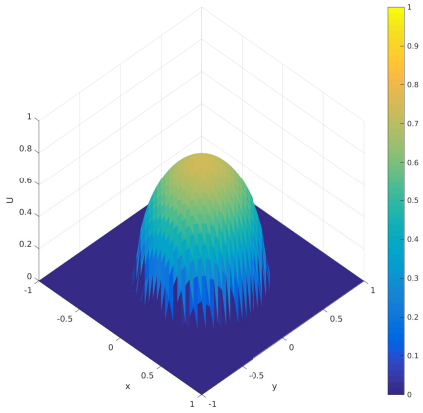
We take $t \in [0, 0.1]$. The numerical results are shown in Fig. 3.17, and they are comparable with those in [14]. □

3.4 Some results for 3D setting

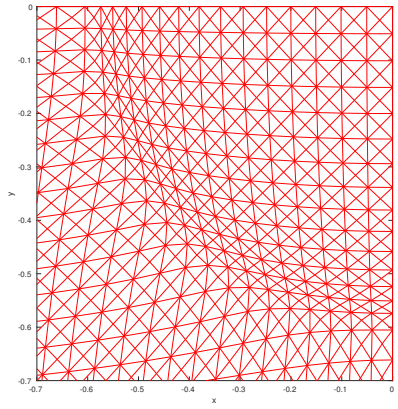
The UE-method of this chapter can be extended to the 3D setting. For example, Fig 3.7 shows a convergence history for the method applied to (3.1) having the Barenblatt-Pattle solutions and equipped with Hessian-based mesh adaptation, for cases of $m = 1, 2$, and 3. Currently, our computing resources only allow a simulation on coarser meshes, for which the method shows a first order convergence. This is nonetheless consistent with the results for 2D, as seen in Fig. 3.4b, as we note that the method there converges with a less-than-optimal order on coarse meshes, whereas the second order convergence happens only on finer meshes.



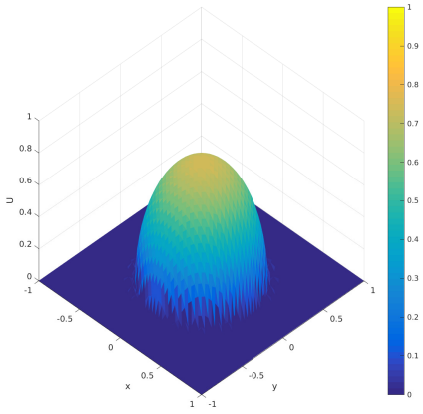
(a) Uniform mesh



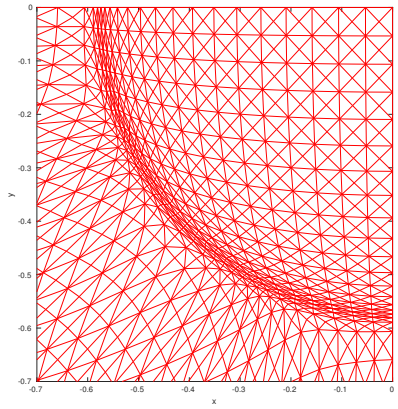
(b) Uniform mesh



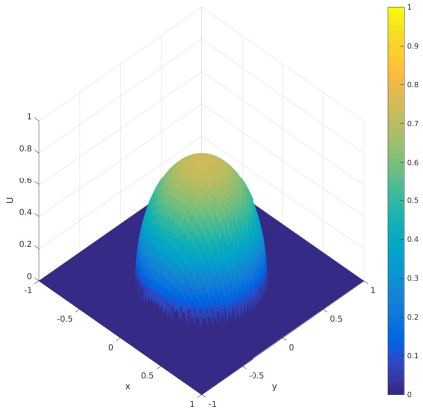
(c) Arclength metric tensor



(d) Arclength metric tensor

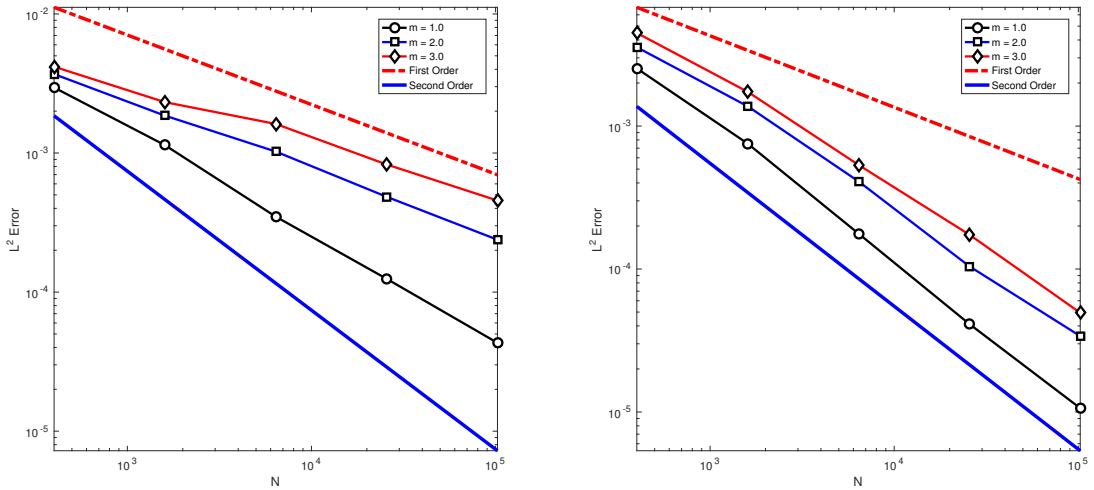


(e) Hessian-based metric tensor



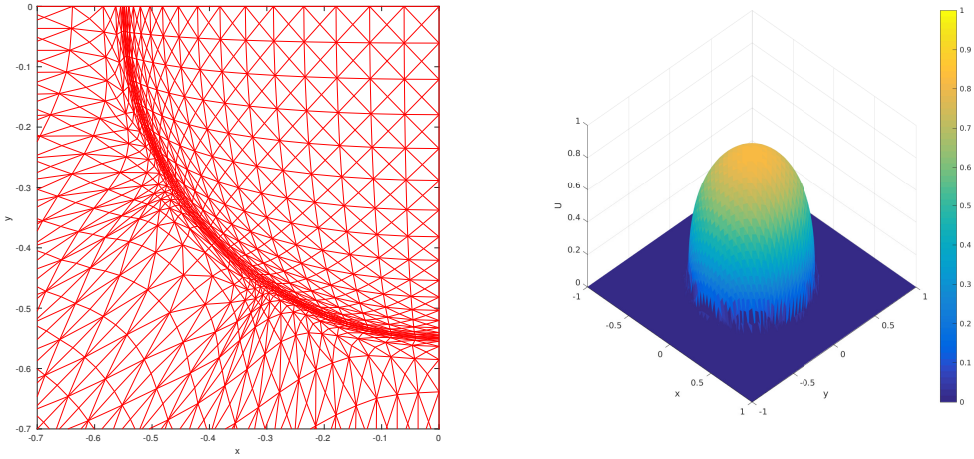
(f) Hessian-based metric tensor

Figure 3.3: The meshes (closer view near (-0.35, -0.35)) and computed solutions at $t = T$ obtained with uniform and arclength- and Hessian-based adaptive meshes ($N = 25600$).



(a) Uniform mesh (b) Hessian-based adaptive mesh

Figure 3.4: Convergence history for the UE-method applied on uniform and Hessian-based meshes.



(a) Mesh (b) Computed solution

Figure 3.5: Example 3.1.1. The final mesh (close view near (-0.35, -0.35)) and computed solution for $m = 3$ with the Hessian-based mesh adaptation ($N = 25600$).

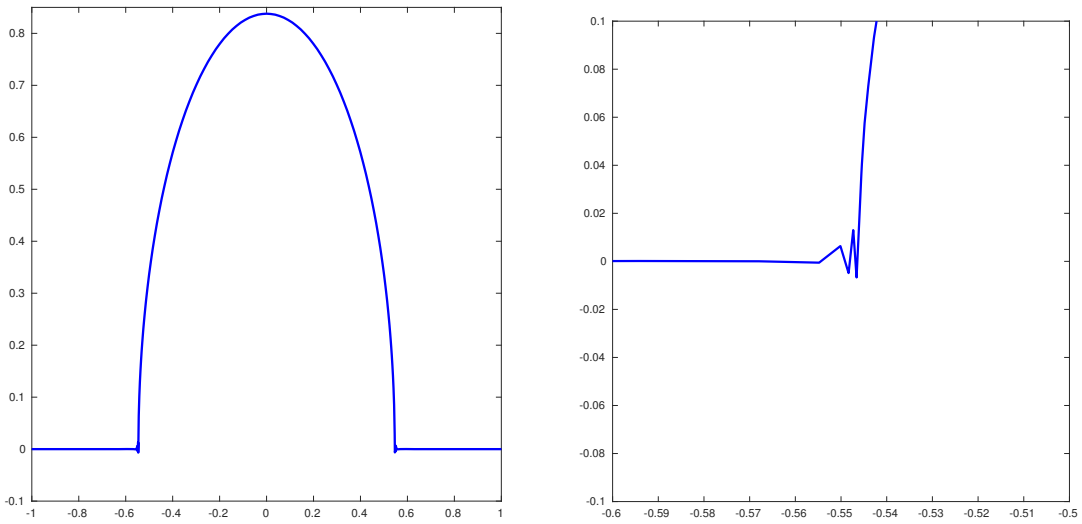


Figure 3.6: Observation of oscillations in the UE-method.

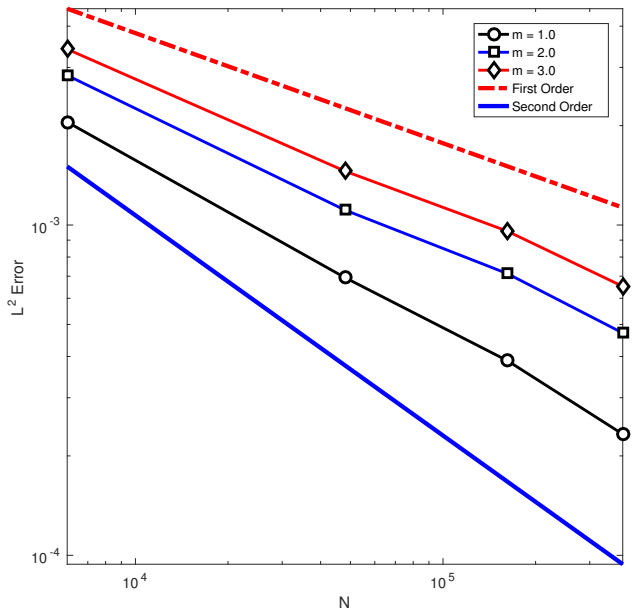
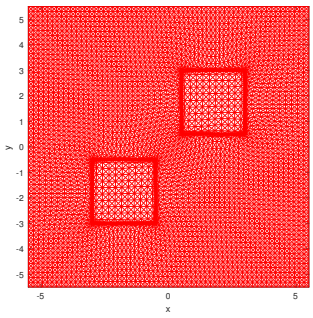
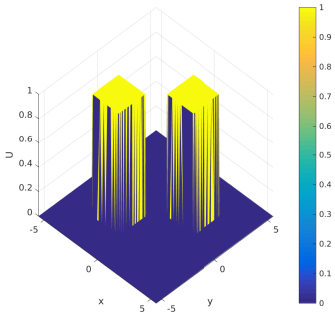


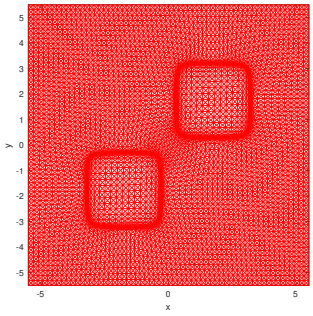
Figure 3.7: Convergence history of the UE-method with Hessian-based adaptivity in 3D.



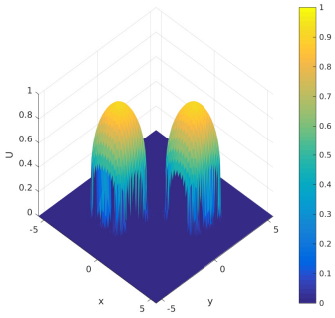
(a) $t = 0$



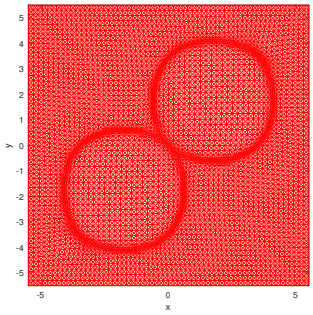
(b) $t = 0$



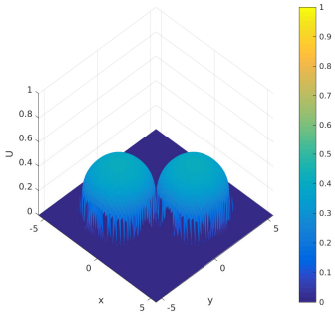
(c) $t = 0.51$



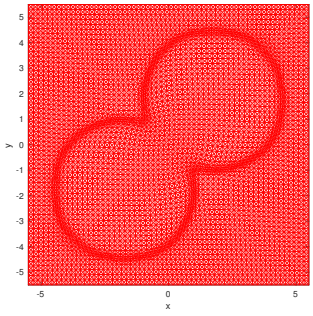
(d) $t = 0.51$



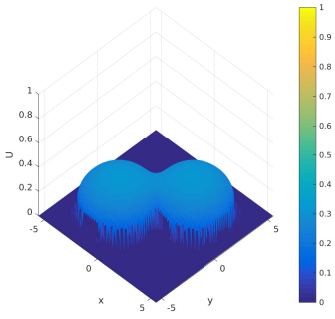
(e) $t = 100.01$



(f) $t = 100.01$

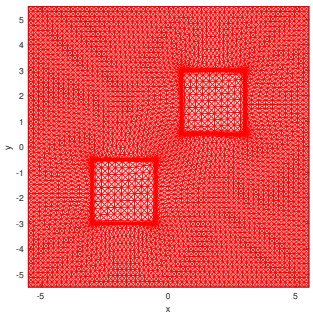


(g) $t = 500$

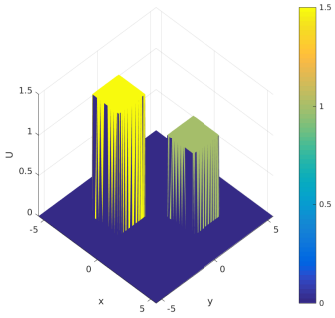


(h) $t = 500$

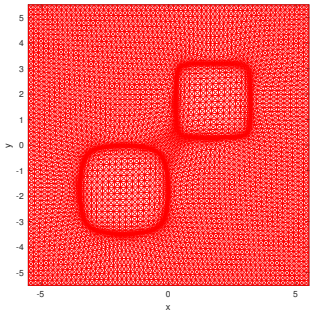
Figure 3.8: Example 3.2.1. An adaptive mesh and the corresponding computed solution at various time instants ($N = 14400$).



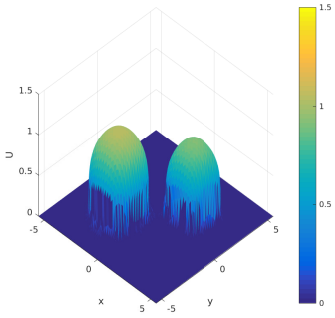
(a) $t = 0$



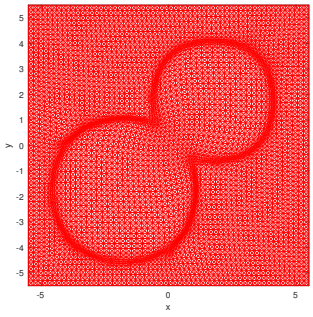
(b) $t = 0$



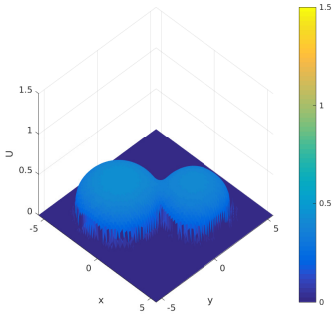
(c) $t = 0.5$



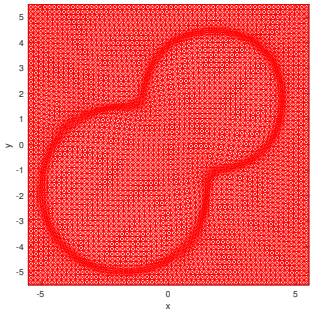
(d) $t = 0.5$



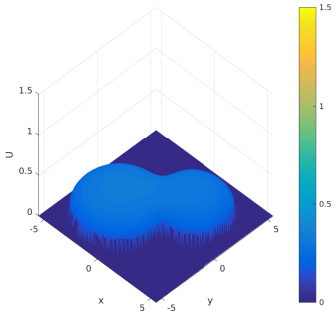
(e) $t = 100$



(f) $t = 100$

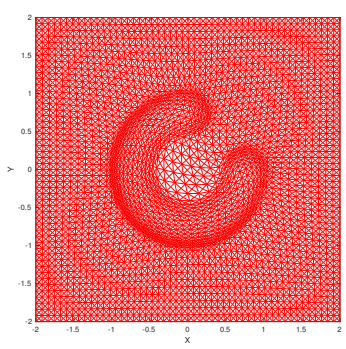


(g) $t = 500$

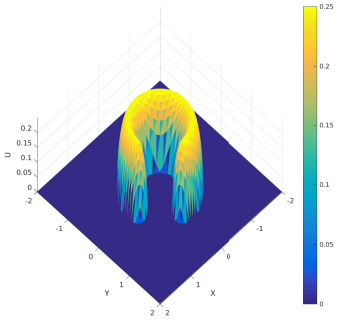


(h) $t = 500$

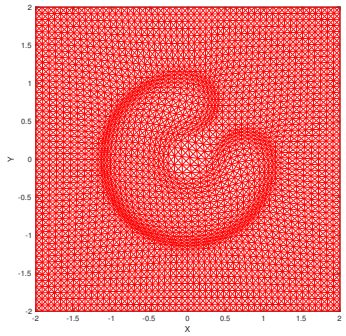
Figure 3.9: Example 3.2.2. An adaptive mesh and the corresponding computed solution at various time instants ($N = 14400$).



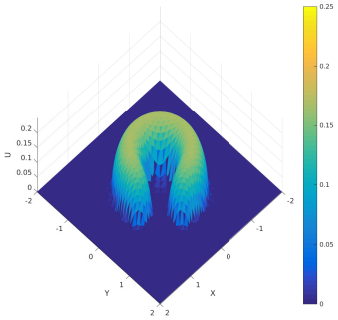
(a) $t = 0$



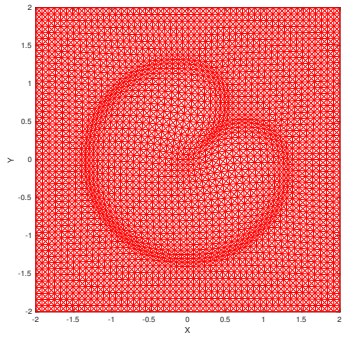
(b) $t = 0$



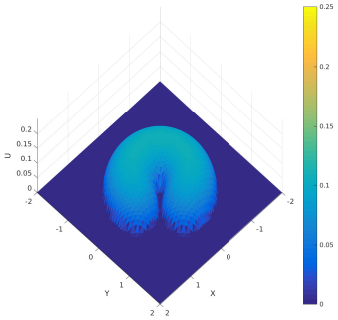
(c) $t = 1$



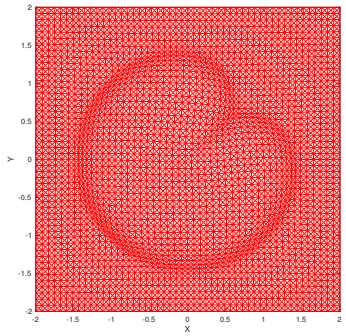
(d) $t = 1$



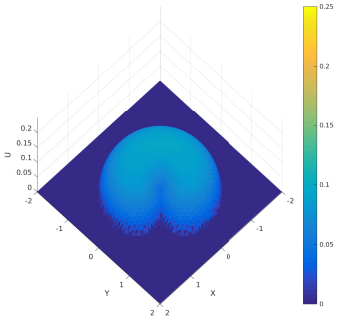
(e) $t = 8$



(f) $t = 8$

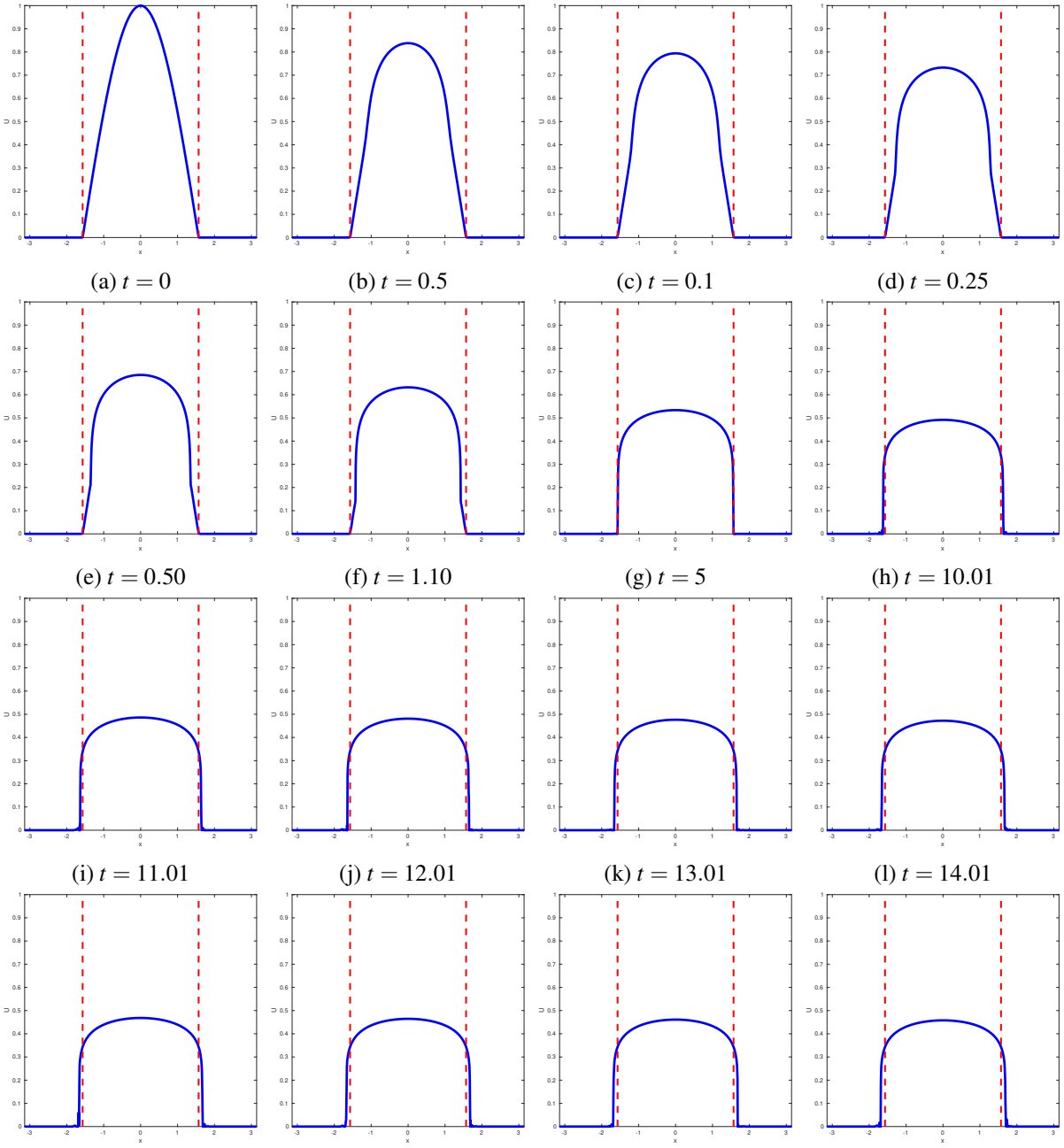


(g) $t = 15$



(h) $t = 15$

Figure 3.10: Example 3.2.3. A computed solution is shown at various time instants ($N = 10000$).



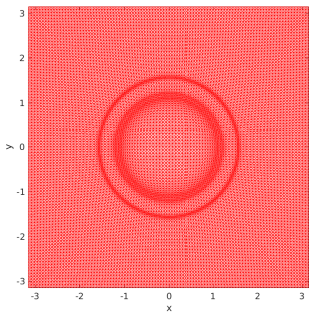
(a) $t = 0$ (b) $t = 0.5$ (c) $t = 0.1$ (d) $t = 0.25$

(e) $t = 0.50$ (f) $t = 1.10$ (g) $t = 5$ (h) $t = 10.01$

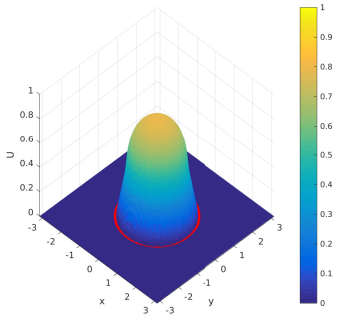
(i) $t = 11.01$ (j) $t = 12.01$ (k) $t = 13.01$ (l) $t = 14.01$

(m) $t = 15.01$ (n) $t = 16.01$ (o) $t = 17.01$ (p) $t = 18.01$

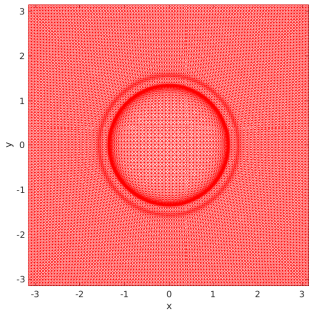
Figure 3.11: Example 3.2.4. The cross section at $y = 0$ of a computed solution is shown at various time instants.



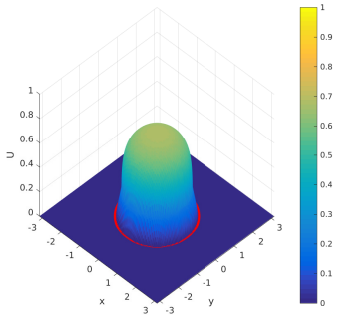
(a) $t = 0.1$



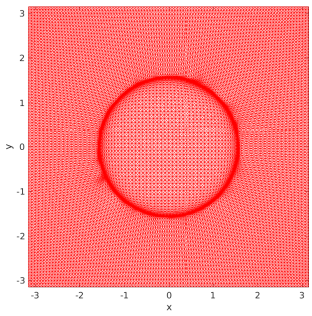
(b) $t = 0.1$



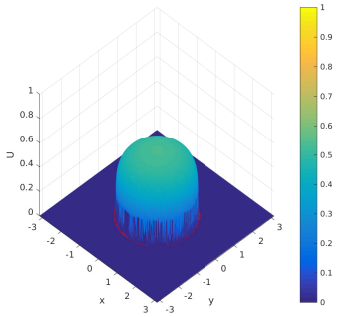
(c) $t = 0.5$



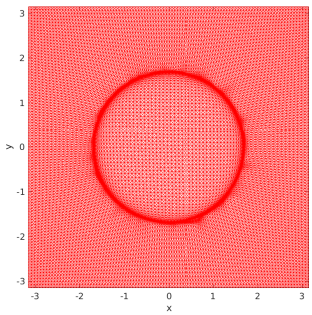
(d) $t = 0.5$



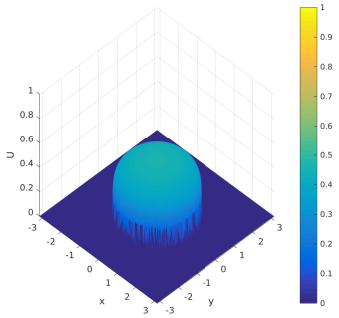
(e) $t = 5$



(f) $t = 5$

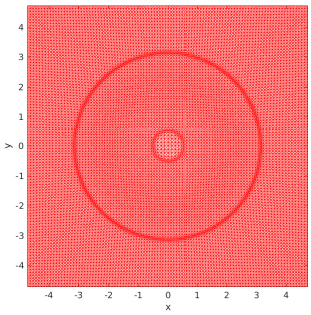


(g) $t = 18.01$

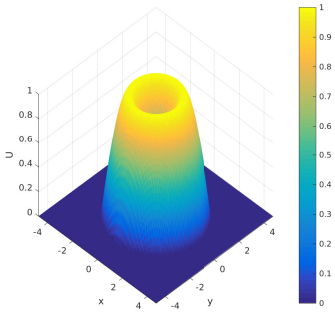


(h) $t = 18.01$

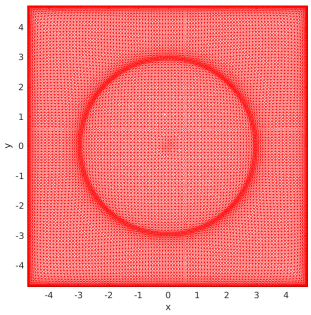
Figure 3.12: Example 3.2.4. A computed solution is shown at various time instants ($N = 40000$).



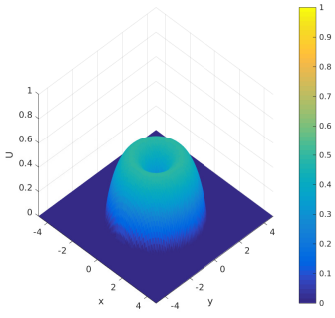
(a) $t = 0$



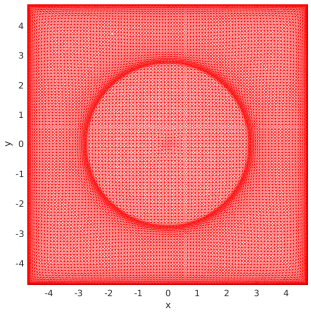
(b) $t = 0$



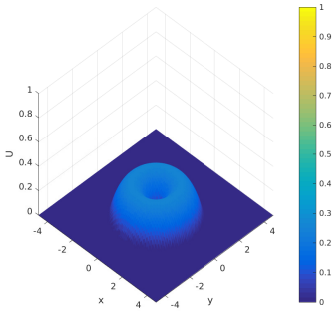
(c) $t = 0.40$



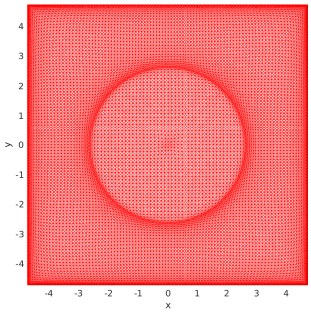
(d) $t = 0.40$



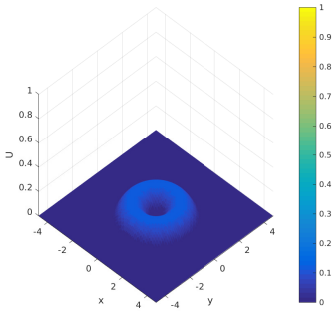
(e) $t = 0.64$



(f) $t = 0.64$

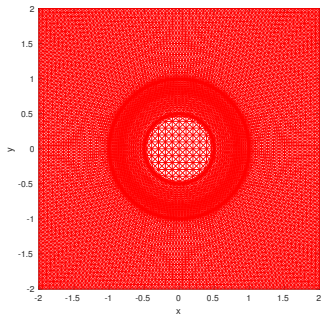


(g) $t = 0.80$

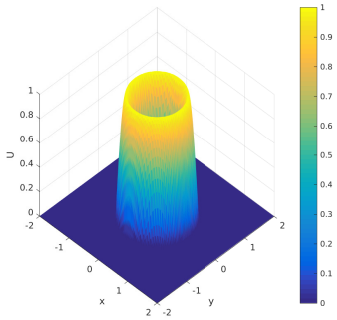


(h) $t = 0.80$

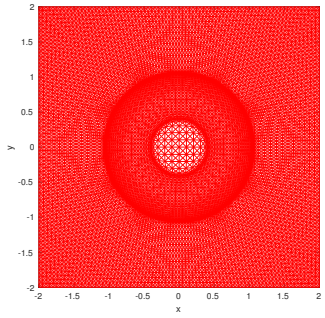
Figure 3.13: Example 3.3.1. An adaptive mesh and the corresponding solution at various time instants ($N = 40000$).



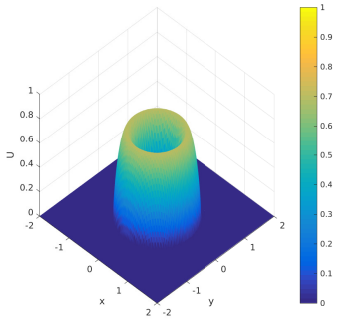
(a) $t = 0$



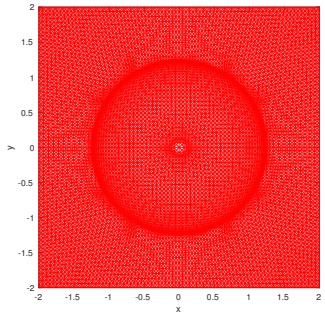
(b) $t = 0$



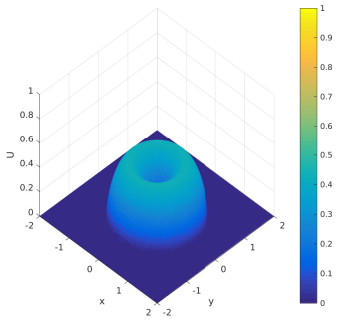
(c) $t = 0.02$



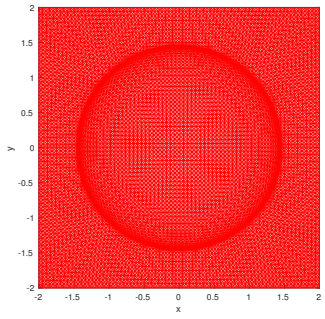
(d) $t = 0.02$



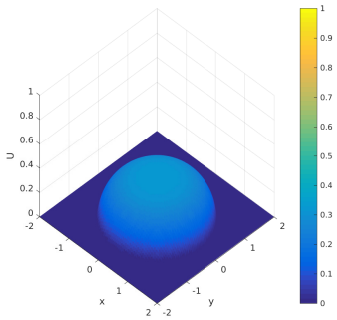
(e) $t = 0.10$



(f) $t = 0.15$

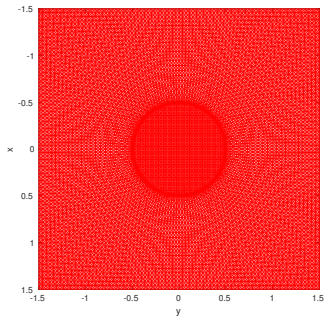


(g) $t = 0.70$

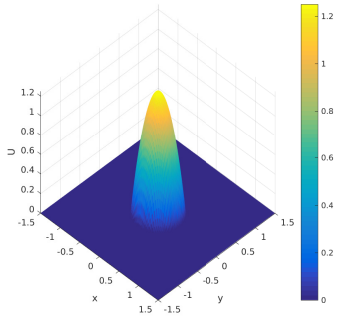


(h) $t = 0.70$

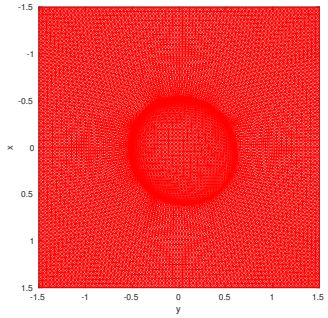
Figure 3.14: Example 3.3.2. An adaptive mesh and the corresponding solution at various time instants ($N = 25600$).



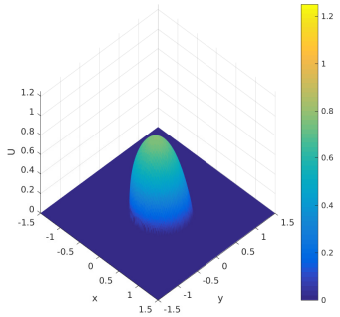
(a) $t = 0$



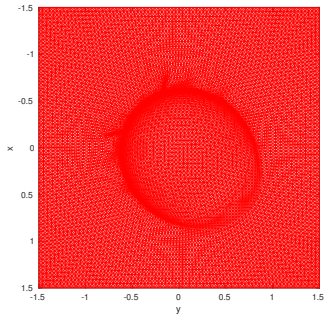
(b) $t = 0$



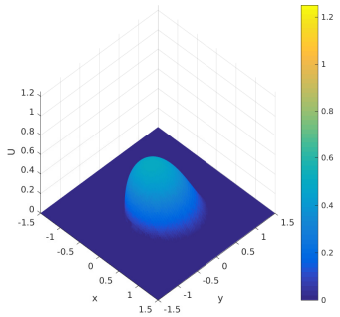
(c) $t = 0.06$



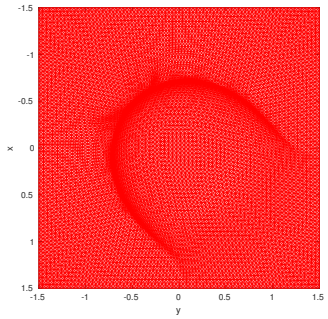
(d) $t = 0.06$



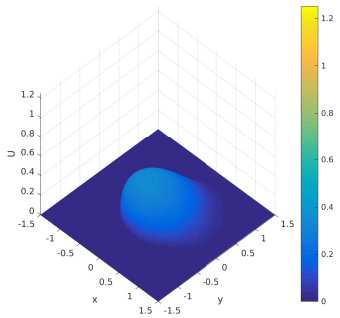
(e) $t = 0.28$



(f) $t = 0.28$

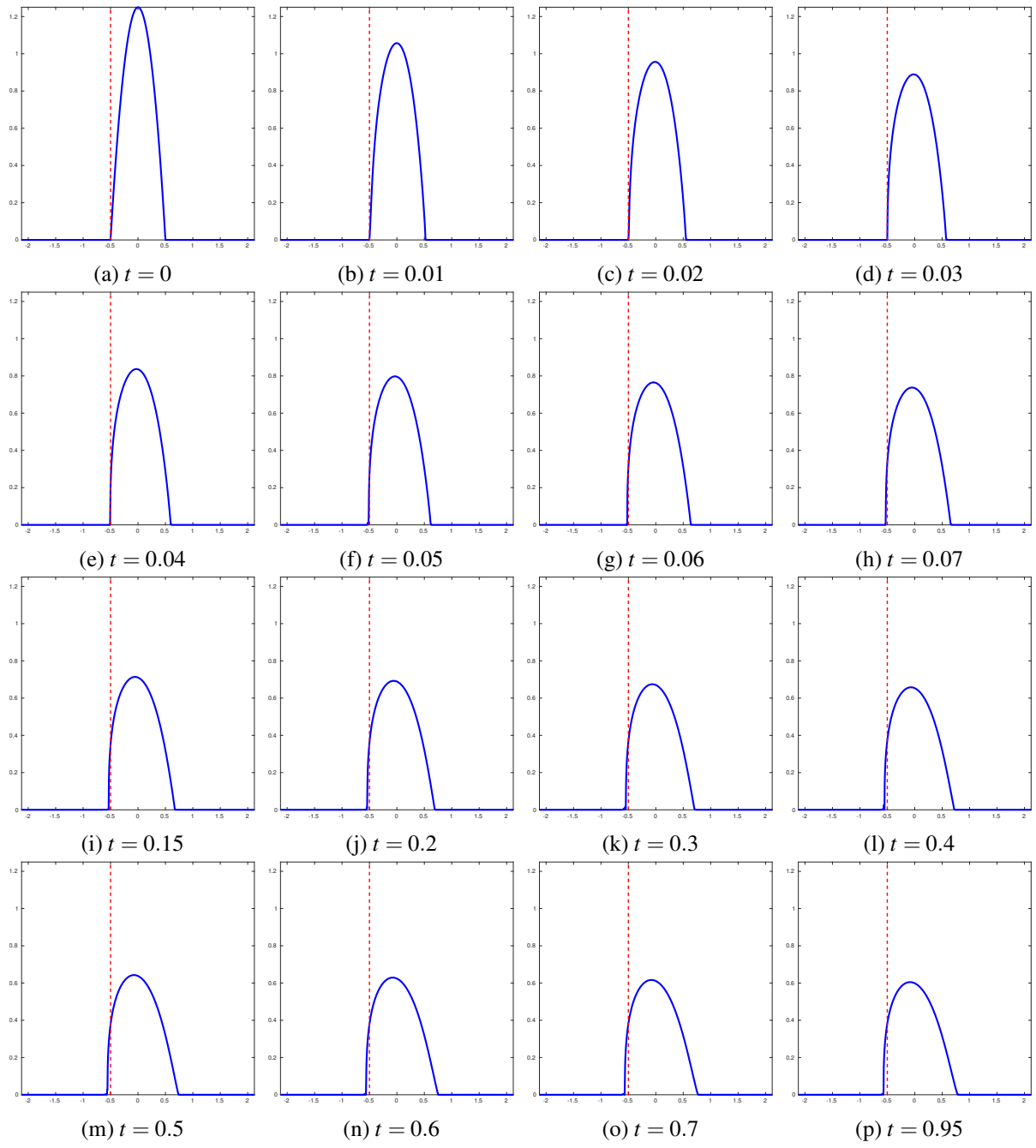


(g) $t = 0.90$



(h) $t = 0.90$

Figure 3.15: Example 3.3.3. An adaptive mesh and the corresponding solution at various time instants ($N = 25600$).



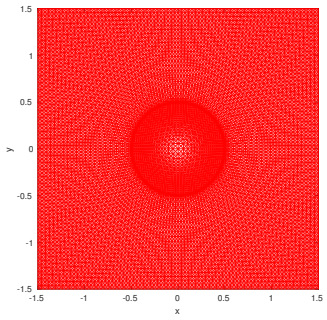
(a) $t = 0$ (b) $t = 0.01$ (c) $t = 0.02$ (d) $t = 0.03$

(e) $t = 0.04$ (f) $t = 0.05$ (g) $t = 0.06$ (h) $t = 0.07$

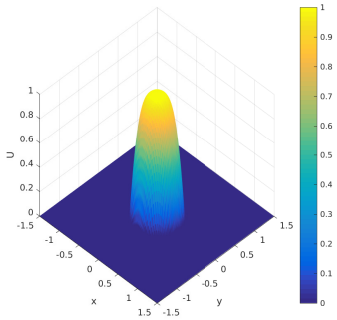
(i) $t = 0.15$ (j) $t = 0.2$ (k) $t = 0.3$ (l) $t = 0.4$

(m) $t = 0.5$ (n) $t = 0.6$ (o) $t = 0.7$ (p) $t = 0.95$

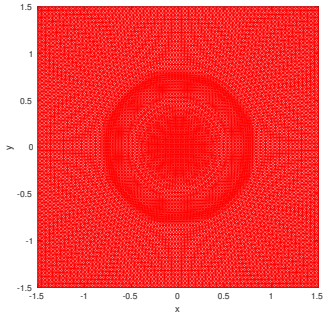
Figure 3.16: Example 3.3.3. The cross section at $y = x$ of a computed solution is shown at various time instants.



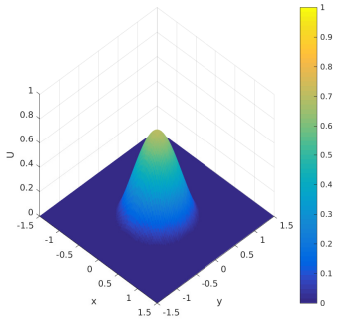
(a) $t = 0.00$



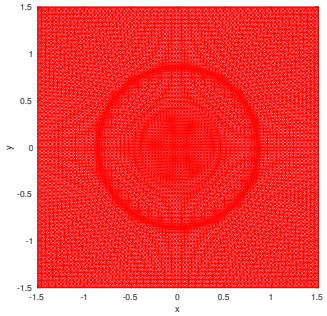
(b) $t = 0$



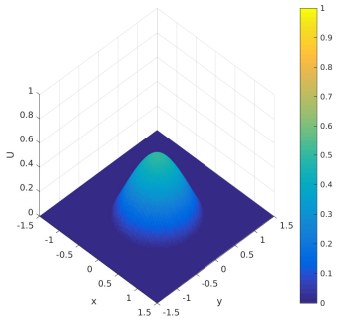
(c) $t = 0.03$



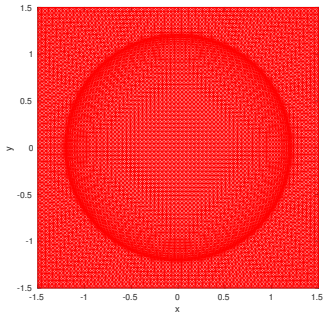
(d) $t = 0.03$



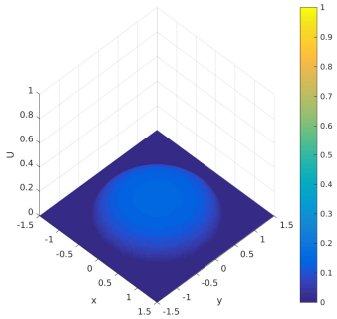
(e) $t = 0.06$



(f) $t = 0.06$



(g) $t = 0.50$



(h) $t = 0.50$

Figure 3.17: Example 3.3.4. An adaptive mesh and the corresponding solution at various time instants ($N = 25600$).

Chapter 4

The U-method: A nonembedding numerical solution for the U-formulation of PME

In Chapter 3, we have presented an embedding moving mesh method for solving the PME (1.2). Even though the method has a few great advantages in terms of its robustness (i.e. applicable to complicated domains) and flexibility (i.e extendable to a more general PME), it also has two major weaknesses. The first one is the loss of regularity as a consequence of extending the solution beyond its natural support. Though our UE-method improves the convergence order via the Hessian-based adaptivity (cf. Fig. 3.4b), it does so by clustering the mesh nodes towards the boundary, which causes some elements near the boundary to be very flat (i.e. with volume almost zero, cf. Fig 3.3). This in effect increases the condition number of the mass matrix $B(\boldsymbol{X})$ in system (2.34), which makes the solution of the DAE more challenging. Another weakness of the embedding method is the higher consumption of computer memory and CPU time due to the requirement of extraneous mesh points outside the solution's support.

In this chapter, we present our "U-method" for solving the IBVP (1.2) by the nonembedding approach (cf. Section 2.5), where "U" stands for the U-formulation of PME (cf. Pg. 1). Since the method only discretizes within the support of solution, it eliminates the need for extraneous mesh points and reduction in the solution's regularity, thus having great potentials in overcoming

the aforementioned disadvantages. We recall from Section 2.5.4 that in order to implement the nonembedding technique, the original IBVP (1.2) needs to be modified into

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u)\,, & \text{in} \quad \Omega(t)\,, \quad t \in (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x})\,, & \text{in} \quad \Omega(t_0) \\ u(\boldsymbol{x}, t) = 0\,, & \text{on} \quad \Gamma(t) := \partial\Omega(t)\,, \quad t \in (t_0, T]\,, \\ \Gamma'(t) = \lim\limits_{\boldsymbol{x} \to \Gamma(t)^-} -\nabla \left(\dfrac{|u|^m(\boldsymbol{x}, t)}{m} \right) \cdot \hat{\boldsymbol{n}} & \text{in} \quad (t_0, T]\,, \end{cases} \tag{4.1}$$

where Darcy's law is included to account for the boundary movements. In order to evaluate the accuracy of our method, we use Barenblatt-Pattle solutions, and measure the error in L^2-norm, i.e.

$$\|u_h - u\|_{L^2(t_0,T;L^2(\tilde{\Omega}))} = \left(\int_{t_0}^{T} \int_{\tilde{\Omega}} (u_h - u)^2(\boldsymbol{x}, t) d\boldsymbol{x} dt \right)^{\frac{1}{2}}, \tag{4.2}$$

where $\tilde{\Omega}$ is a fixed domain containing $\Omega(T)$. For these solutions, unless otherwise stated, we will let the final time be

$$T = \frac{0.1 + t_0}{2}\,,$$

where t_0 is given in (2.15).

Before presenting the main numerical results, we would like to perform some numerical experiments and make observations regarding the following components of our method:

1. $\underline{\tau}$: We would like to see how τ (cf. Subsection 2.5.2, pg. 36), a parameter for specifying how fast the mesh movement will react to any change in the metric tensor $\mathbb{M}$ (cf. Subsection 2.5.2, Pg. 33), affects the mesh movements and the overall accuracy of the solution.

2. $\underline{\Delta t_{max}}$: We seek to observe how the size of the largest time step

$$\Delta t_{max} = \max_{n=1,\ldots,n_f} (t_n - t_{n-1})\,, \tag{4.3}$$

allowed in our method affects the solution.

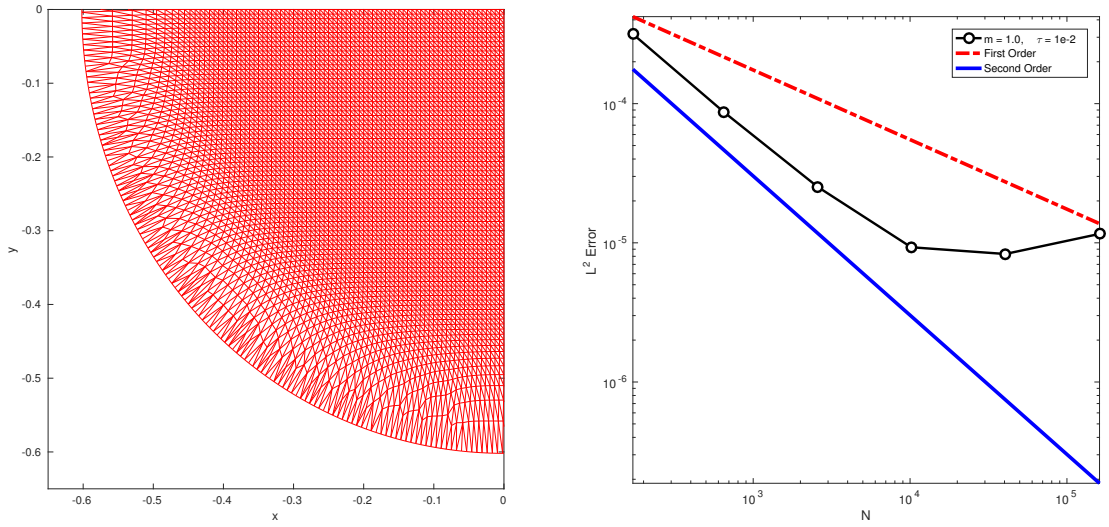
3. $\underline{\mathbb{M}(\boldsymbol{x})}$: We would like to compare between the meshing strategies which are specified by different choices of the metric tensor $\mathbb{M}$, as seen in (2.36), (2.37), and (2.38) of subsection 2.5.2. Namely, we would like to understand how the solution is affected if simulated on a uniform mesh, and on a mesh generated under the arclength-based or Hessian-based adaptivity.

4.1 Effect of τ

As indicated in subsection 2.5.2 (pg. 36), the value of the time-scaling parameter τ dictates how fast the mesh points will move in regard to changes in solution, in the metric tensor $\mathbb{M}$, and in the size of the solution's support.

We begin applying our numerical method for (4.1) where $m = 1$, with a rather standard choice of $\tau = 10^{-2}$, where $\mathbb{M} = \mathbb{I}$, and the free boundary is given exactly at each time level according to the Barenblatt-Pattle solution (2.14). However, some difficulties have arisen with this choice of τ. First, we observe that the method's convergence order, though starting out with a second-order rate for coarser meshes, declines very quickly as the meshes get finer (Fig. 4.1b). We then decide to take a closer look at a representative mesh in one particular experiment, and observe that the distance between the boundary nodes to their nearest interior nodes are much greater than the diameter of any interior element (one with no vertices on the boundary). Hence, with this typical value for τ, the interior vertices could not keep up with the movement of the free boundary; this motivates us to use a smaller value for τ. Indeed, using a faster scaling of mesh movement by setting, for example $\tau = 10^{-4}$, causes the mesh to remain uniform throughout the computation (See the mesh in Fig. 4.2a), which, as opposed to the former case, produces a better convergence result (Fig. 4.2b). This investigation suggests that a smaller value of τ is more conducive for the free boundary problem, which brings about a better overall accuracy. Besides the option of using a generic value e.g. $\tau = 10^{-4}$ or $\tau = 10^{-5}$, we can make this parameter adaptive to the mesh, such as

$$\tau = \min\left\{10^{-3}, \frac{10^{-1}}{N}\right\}. \tag{4.4}$$



(a) Final mesh

(b) Convergence history

Figure 4.1: Effects of $\tau = 10^{-2}$ on mesh movement and accuracy for the U-method.

This suggestion is based on the convergence history in Fig. 4.1b, where the convergence order declines gradually as the mesh gets finer. For the rest of the chapter, unless otherwise noticed, we will use formula (4.4) as a standard choice for the parameter τ of our method.

4.2 Effect of Δt_{max}

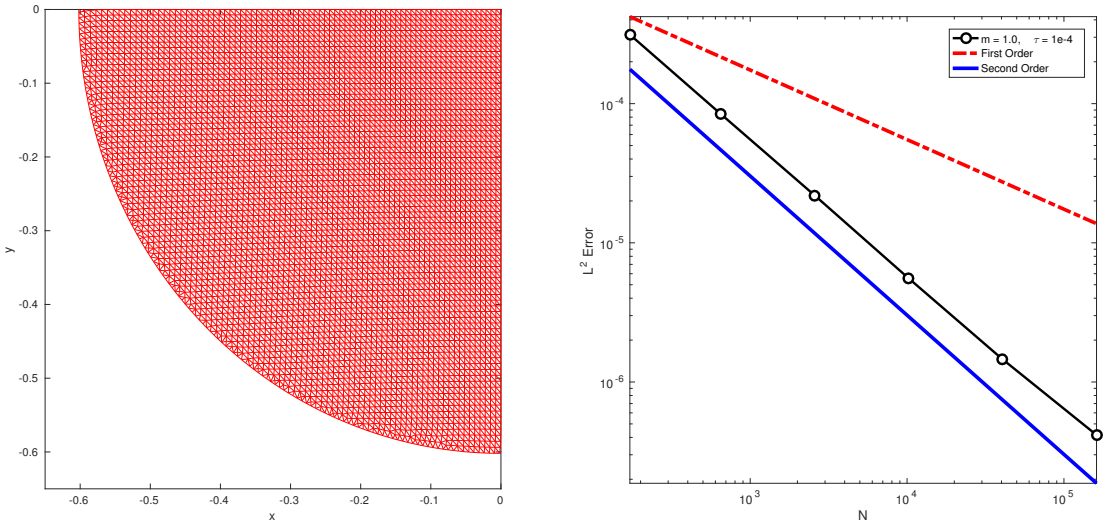
At this current stage of our research, Darcy's law

$$\Gamma'(t) = \lim_{\boldsymbol{x}\to\Gamma(t)^-} -\nabla\left(\frac{|u|^m(\boldsymbol{x},t)}{m}\right)\cdot\hat{\boldsymbol{n}}, \quad \text{in} \quad (t_0,T]$$

is splitted from the bigger system (See the chart in (2.48) of Section 2.5.4), and is discretized and computed, e.g. at time $t = t_n$, by the simple forward Euler scheme given by (cf. Pg. 39)

$$\frac{\Gamma^{n+1}-\Gamma^n}{t_{n+1}-t_n} = -\nabla_h\left(\frac{|u_h^n|^m}{m}\right)\cdot\hat{\boldsymbol{n}}, \tag{4.5}$$

where $\nabla_h(\cdot)$ represents an approximation of $\lim_{\boldsymbol{x}\to\Gamma(t_n)^-}\nabla(\cdot)$ at $t = t_n$, and $\hat{\boldsymbol{n}}$ denotes the unit outward normal to the boundary (See Fig. 2.6). We find through a straightforward analysis that this



(a) Final mesh

(b) Convergence history

Figure 4.2: Effects of $\tau = 10^{-4}$ on mesh movement and accuracy for the U-method.

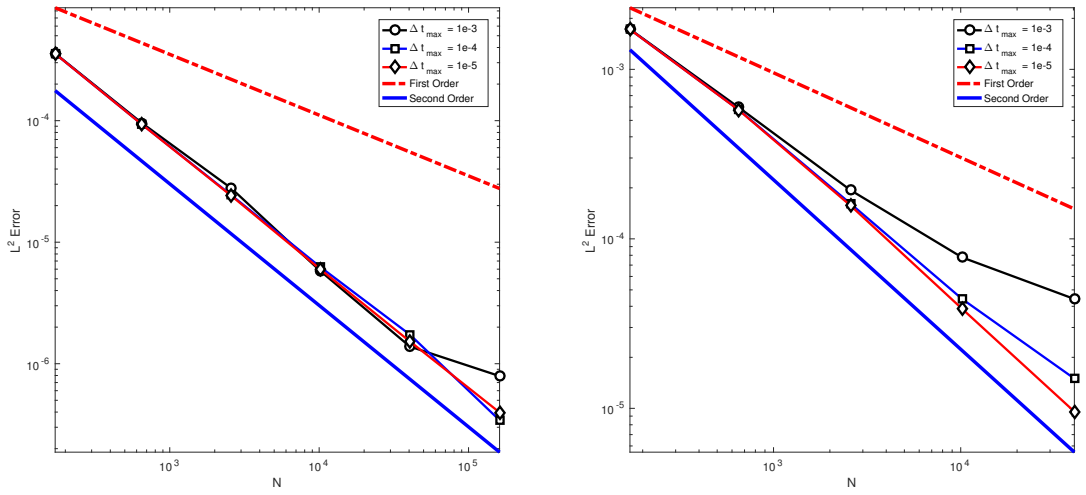
splitting technique indeed introduces the error terms

$$\mathscr{O}(h^2) + \mathscr{O}(\Delta t_{max})\mathscr{O}(h) \tag{4.6}$$

into the numerical solution, where h is the diameter of the largest element of $\mathscr{T}_h$ and Δt_{max} is the largest time step allowed (as defined in (4.3)). This implies that if we move the boundary points via Darcy's law with such scheme, the convergence order deteriorates for larger maximum time step for very fine meshes. This implies the necessity of using a smaller time step.

For a confirmation, we again consider the Barenblatt-Pattle solution and apply our method in several experiments where the maximum time steps are no greater than $10^{-3}, 10^{-4}$, and 10^{-5}, respectively. Again, we use a uniform mesh with boundary points given exactly at each time level, and other parameters chosen appropriately. Fig. 4.3 shows the convergence histories for cases $m = 1$ and $m = 2$. It demonstrates that too big a time step reduces the accuracy and order of convergence for very fine meshes, while a sufficiently small time step gives an optimal (second-order) convergence behavior.

Remark 4.2.1. We might improve the method's independence on the size of average time step through using a solution predictor at each iteration. Namely, before performing the iteration, with



(a) $m = 1$ (b) $m = 2$

Figure 4.3: Convergence history of the method for various fixed maximal time steps (original formulation).

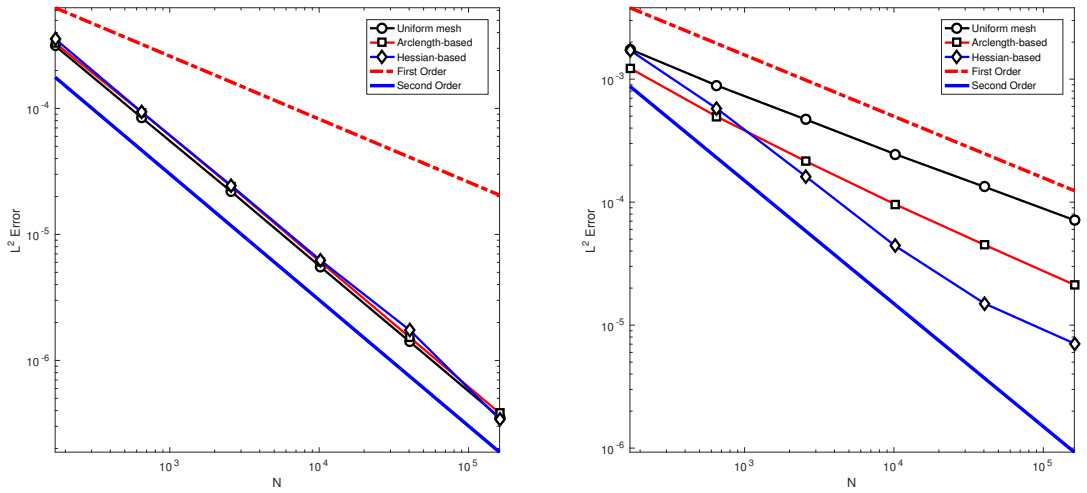
the mesh being fixed, we solve for the solution at the next time level, and use this solution to aide with the estimation of the next solution on the moving mesh.

4.3 Effect of mesh adaptation strategies

In this section, we would like to observe how different mesh adaptation strategies affect our numerical solution. Recall from subsection 2.5.2 that we have three major choices of the metric tensor $\mathbb{M}$, given in equations (2.36), (2.37), and (2.38), by which the MMPDE method generates a uniform mesh, an arclength-based adaptive mesh, and a Hessian-based adaptive mesh, respectively.

In order to observe the performance of each of these meshing strategies, we apply our method, with appropriately chosen parameters, to the IBVP (4.1), where exact Barenblatt-Pattle solutions are known. Again, for simplicity, the boundary at each time level is given exactly (with formula (2.14)), so that our evaluation of these strategies will not be unnecessarily interfered with boundary error.

As one may expect, the U-method applied with an adaptive strategy is more accurate than the same on an uniform mesh. Moreover, between the two approaches of adaptivity, the Hessian-



(a) $m = 1$ (b) $m = 2$

Figure 4.4: Convergence history of the U-method for the three meshing strategies.

based strategy produces, in most situations, better results than the arclength-based one (See Fig. 4.4). The exception is for the case $m = 1$ and other situations where the solution does not have a steep front. In such cases, the performance of these three approaches is similar to one another (See Fig. 4.4a). On the other hand, for the case $m = 2$, where the solution has a steep front (i.e. the slope tends to infinity as one approaches the boundary from the inside), there is a significant deviation in performance between these there approaches (See Fig. 4.4b). For the nonadaptive (uniform mesh) approach, the convergence order is less than one, which is undesirable. For the arclength-based approach, we observe a first-order convergence, which is still less than the optimal rate. On the other hand, for the Hessian-based method, there is a major breakthrough, as a convergence of second order, which is optimal for a linear FE method, is observed providing that the mesh is not so fine (it is possible that a smaller time step will be required for achieving a second-order convergence over finer meshes).

Remark 4.3.1. From the results of subsections 4.1, 4.2, and 4.3, we observe that in order to apply our method effectively to find a numerical solution for (4.1), it is practical that we use

1. A sufficiently small value for τ (e.g. $\tau = 10^{-4}$, 10^{-5}, or the adaptive formula in (4.4)),

2. A sufficiently small Δt_{max} between the time levels, and

3. The adaptive meshing strategy based on the solution's Hessian.

4.4 Effect of PME parameter m on the solution

We know from some error estimates (e.g. (1.3), (1.4), (1.6), and (1.7)) in the Introduction (Chapter 1) that for various numerical methods of PME on a fixed mesh, the convergence order deteriorates as parameter m gets higher. Hence, it is imperative to observe and understand the effect of this parameter on the accuracy of our method. For this purpose, we will again use the Barenblatt-Pattle solution (2.14) to evaluate the accuracy, where it can be verified analytically that the higher the value of m is, the steeper the slope of the Barenblatt-Pattle solution will be near/at its free boundary. Indeed, for $m > 1$, the slope of the solution at the free boundary is infinite, whereas for the case $m = 1$, it is finite. For these reasons, we should expect that the higher m is, the more challenging it will be for numerical simulation. For a confirmation, we apply our U-method to (4.1) and choose (cf. Remark 4.3.1)

$$\tau = 10^{-5}, \quad \Delta t_{max} = 10^{-5}, \quad \text{and} \quad \text{Hessian-based mesh adaptation}$$

for four cases

$$m = 1, \quad m = 1.5, \quad m = 2, \quad \text{and} \quad m = 3.$$

Moreover, we again prescribe the boundary points exactly according to the formula in (2.14) for a more reliable assessment. A convergence history is given in Fig. 4.5a. We observe that for a sufficiently small time step (e.g in this case, $\Delta t_{max} = 10^{-5}$) the accuracy of our method does not depend on parameter m. Even though the magnitude of the error increases as m increases, the method exhibits a second-order convergence for each of these four cases.

4.5 Numerical examples

In this section, we present some numerical results of our U-method as applied to problem (4.1). For each of these experiments, the boundary is computed numerically via Darcy's law, with the scheme given by (4.5). Moreover, following remark 4.3.1, we will use our method with the choice of

$$\tau = \min\left\{10^{-3}, \frac{10^{-1}}{N}\right\}, \quad \Delta t_{max} = 10^{-4},$$

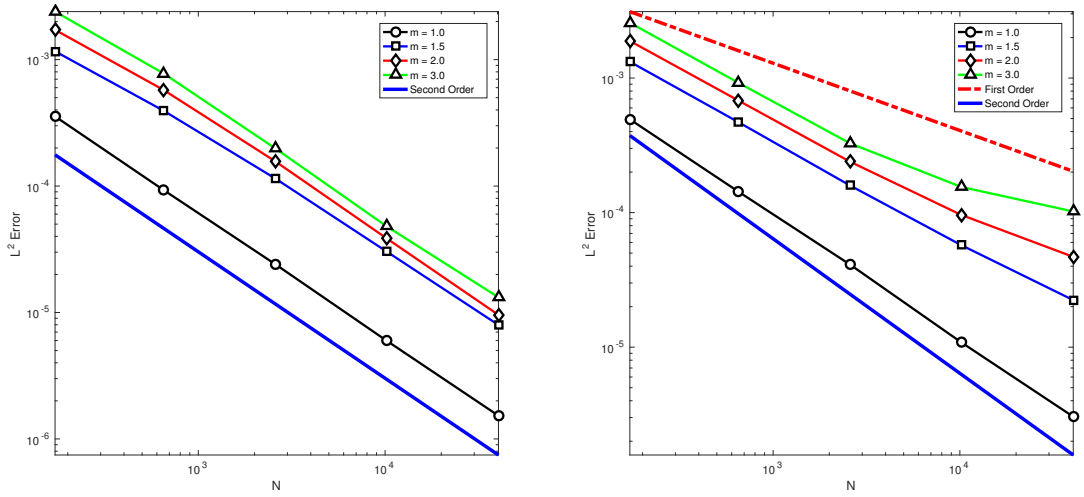
and a Hessian-based mesh adaptation.

Example 4.5.1 (Barenblatt-Pattle solution). We apply our nonembedding moving mesh U-method to problem (4.1) where the initial solution is the Barenblatt-Pattle solution (2.14). Fig. 4.5b shows convergence histories for cases of $m = 1, 1.5, 2$, and 3. Notice that due to the presence of boundary errors from the discretization of Darcy's law, the accuracy of the method is slightly reduced, with order of convergence gradually decreases as m increases. This is in contrast with the scenario where the boundary points are given exactly by (2.14) (see Fig. 4.5a), by which the method produces more accurate results and maintains consistently a second-order convergence. Thus, we see here that a major key for a precise simulation of PME is a precise tracing its free boundary. More research is much needed in this direction.

In Fig. 4.6 and 4.7, we show plots of a representative mesh and its corresponding numerical solution for the case $m = 3$. Notice that the slope of the solution is very steep near the boundary, and also that the mesh is quite anisotropic (i.e. having almost flat elements), as seen through a close-up view in Fig. 4.6b.

Example 4.5.2 (Waiting-time phenomenon). Similar to the simulation of the waiting time phenomenon of the UE-method in Chapter 3 (see Example 3.2.4), we present an analogous experiment by applying the U-method to (4.1) where $m = 3$, and the initial solution is given by (cf. (3.8))

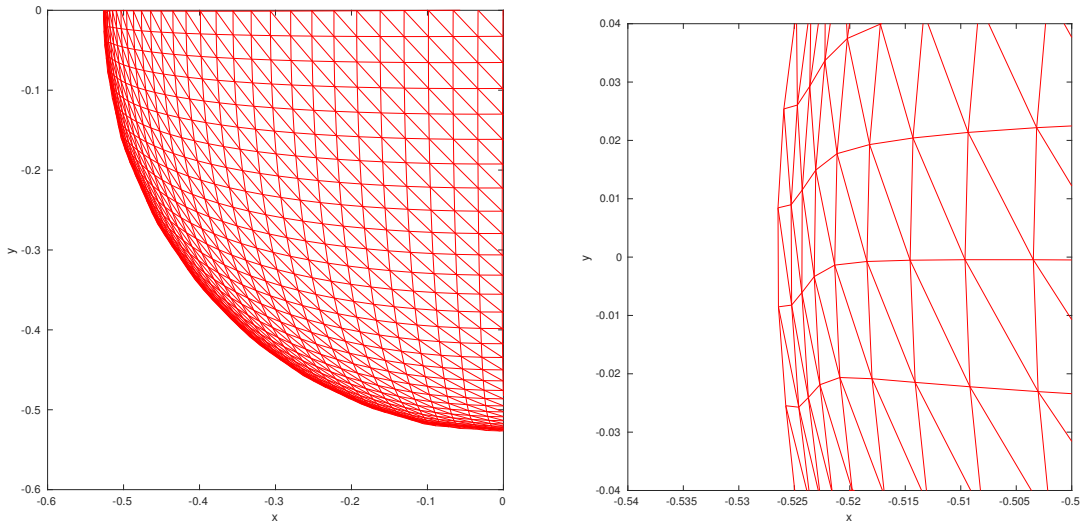
$$u_0(x,y) = \begin{cases} \cos(\sqrt{x^2+y^2}), & \text{for} \quad \sqrt{x^2+y^2} \le \frac{\pi}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$



(a) Computation with exact boundary

(b) Computation with numerical boundary

Figure 4.5: Convergence history of the U-method.



(a) Hessian-based adaptive mesh

(b) A closer look at the boundary

Figure 4.6: Plot of a representative mesh of the U-method for $m = 3$ ($N = 5763$).

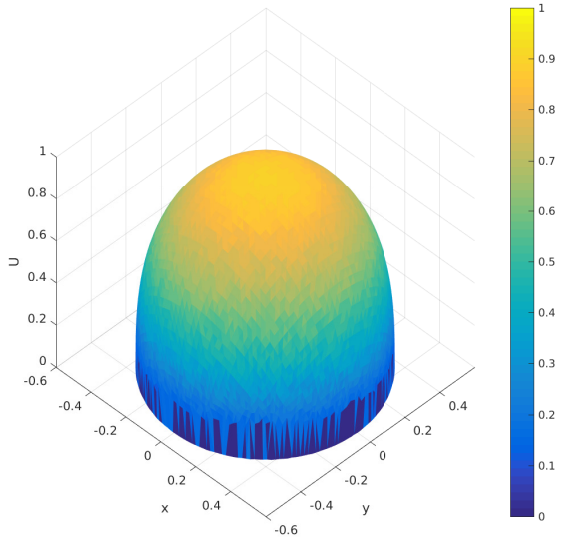


Figure 4.7: Plot of a representative numerical solution by the U-method for $m = 3$ ($N = 5763$).

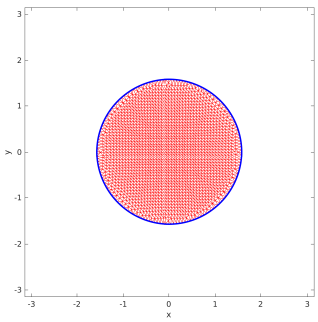
In Fig. 4.8, we present the representative meshes and their corresponding numerical solutions. A reference circle is drawn for each plot of solution to indicate the initial boundary, which verifies the waiting time. For a more detailed verification, we also plot the cross section in the plane $y = 0$ of the solution at various time instants in Fig. 4.9. A careful look suggests that the boundary starts moving at around $t = 1.4$. □

Example 4.5.3 (Complex domain). In this example, we apply our U-method to (4.1) with $m = 1$ and

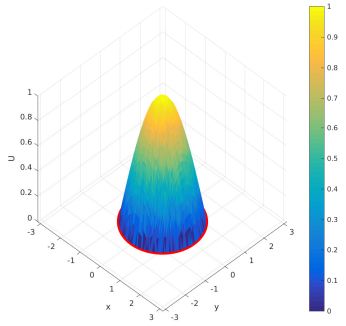
$$u_0(x,y) = \begin{cases} 25\left[0.25^2 - (\sqrt{x^2+y^2} - 0.75)^2\right]^{\frac{3}{2}}, & \text{for} \quad \sqrt{x^2+y^2} \in [0.5,1] \text{ and } (x<0 \text{ or } y<0) \\ 25\left[0.25^2 - x^2 - (y-0.75)^2\right]^{\frac{3}{2}}, & \text{for} \quad x^2+(y-0.75)^2 \le 0.25^2 \text{ and } x \ge 0 \\ 25\left[0.25^2 - (x-0.75)^2 - y^2\right]^{\frac{3}{2}}, & \text{for} \quad (x-0.75)^2+y^2 \le 0.25^2 \text{ and } y \ge 0 \\ 0, & \text{otherwise.} \end{cases}$$

The partial donut-shaped support pertaining to this initial solution is first seen in the paper of Baines et al. [5]. Fig. 4.10 shows the mesh and its corresponding solution at selected time instants.

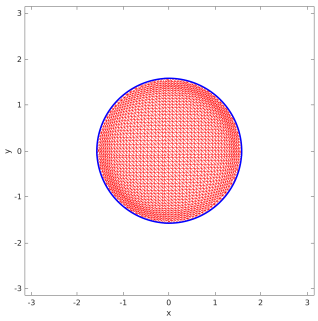
Since the solution is smooth over its support, we have used the boundary-based adaptivity for the MMPDE method, with the choice of $\mathbb{M}$ given in (2.39), where $\alpha = 10^{-6}$ and $r = \frac{1}{2}$, in order to focus more mesh points toward the boundary. We observe that the mesh's evolution is comparable to the results in [5]. □



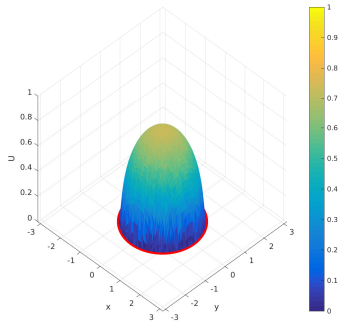
(a) $t = 0$



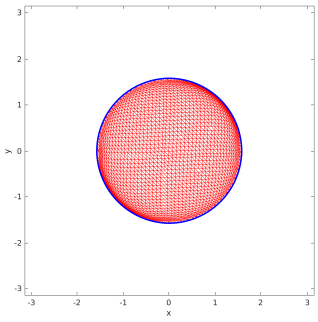
(b) $t = 0$



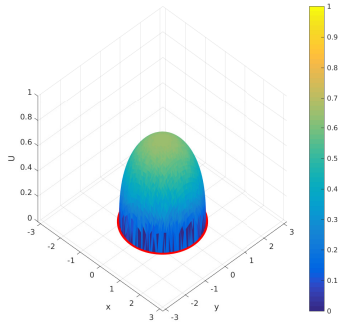
(c) $t = 0.5$



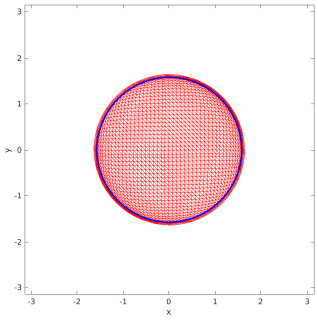
(d) $t = 0.5$



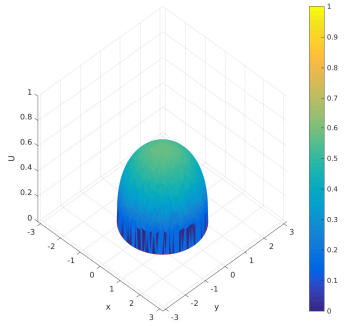
(e) $t = 1$



(f) $t = 1$

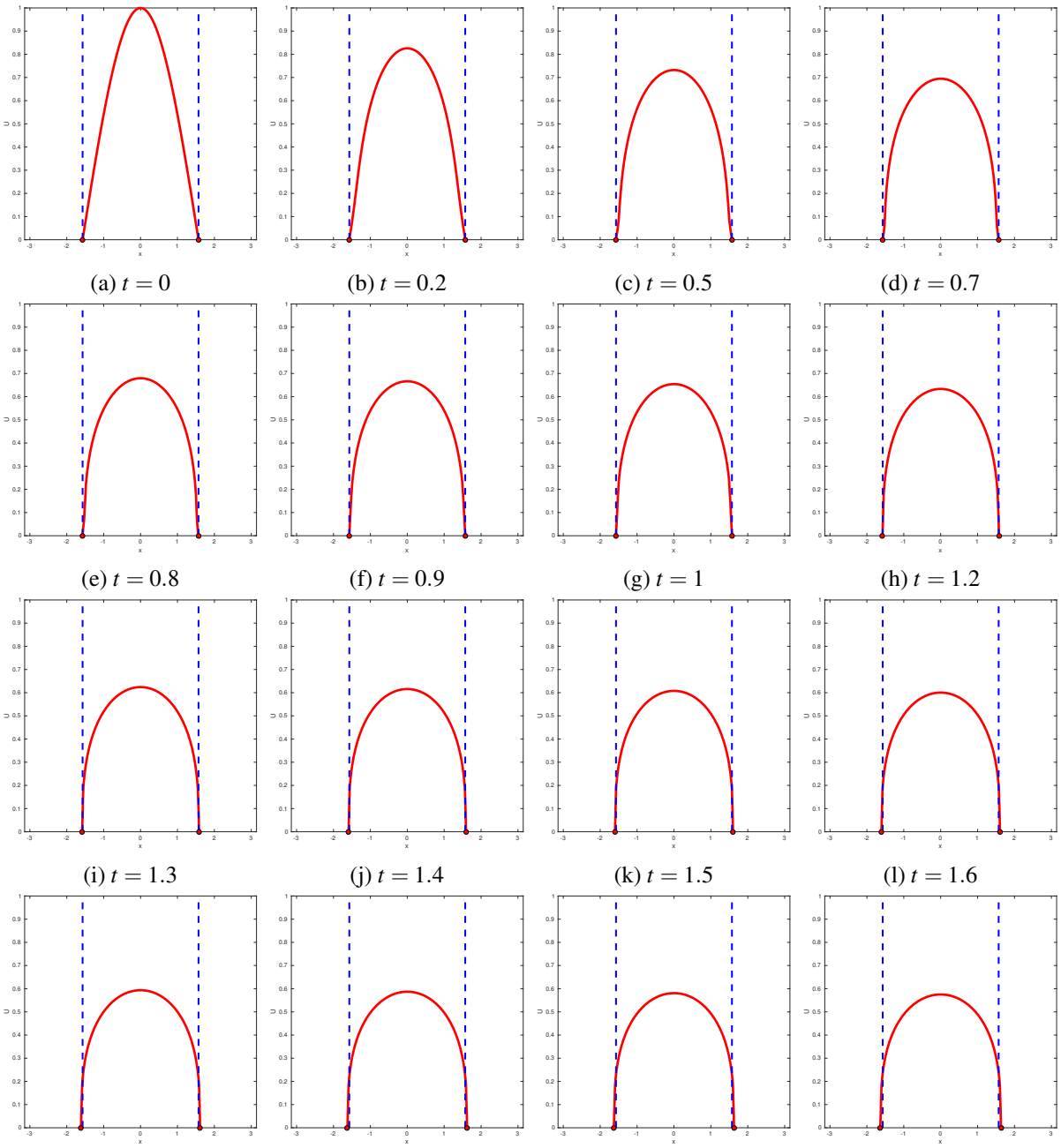


(g) $t = 2$



(h) $t = 2$

Figure 4.8: Example 4.5.2. A computed solution is shown at various time instants ($N = 3743$).



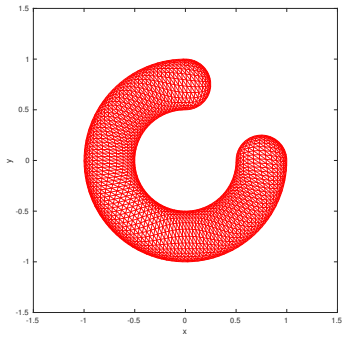
(a) $t = 0$ (b) $t = 0.2$ (c) $t = 0.5$ (d) $t = 0.7$

(e) $t = 0.8$ (f) $t = 0.9$ (g) $t = 1$ (h) $t = 1.2$

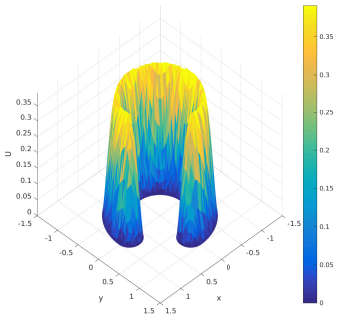
(i) $t = 1.3$ (j) $t = 1.4$ (k) $t = 1.5$ (l) $t = 1.6$

(m) $t = 1.7$ (n) $t = 1.8$ (o) $t = 1.9$ (p) $t = 2$

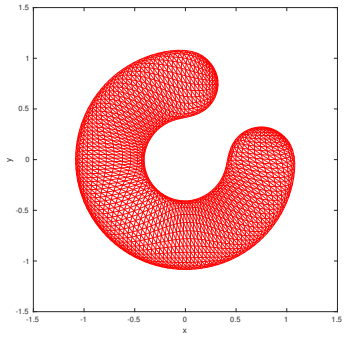
Figure 4.9: Example 4.5.2. The cross section at $y = 0$ of a computed solution is shown at various time instants.



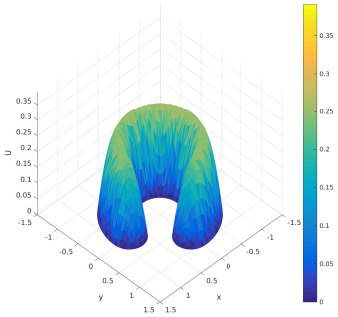
(a) $t = 0.1$



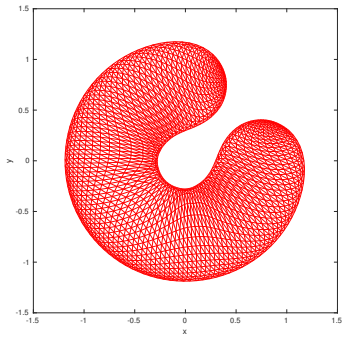
(b) $t = 0.1$



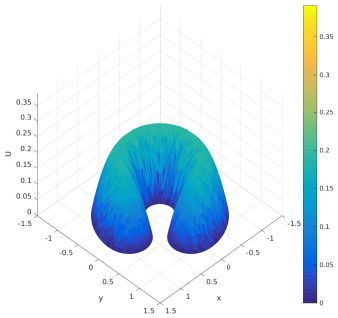
(c) $t = 0.15$



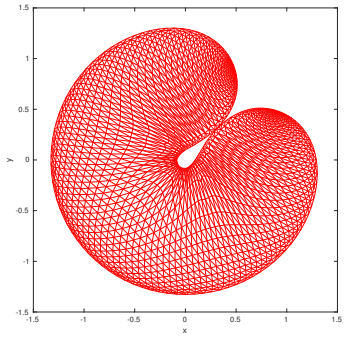
(d) $t = 0.15$



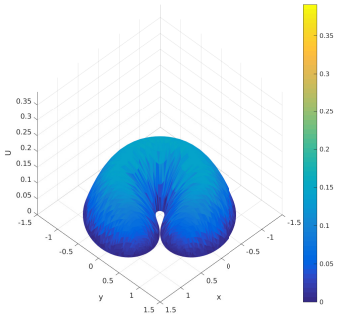
(e) $t = 0.25$



(f) $t = 0.25$



(g) $t = 0.49$



(h) $t = 0.49$

Figure 4.10: Example 4.5.3. A computed solution is shown at various time instants ($N = 3242$).

Chapter 5

The V-method: A nonembedding numerical solution for the V-formulation of PME

In Chapter 4, we have developed a nonembedding adaptive moving mesh method (the U-method) for IBVP (1.2) in order to resolve some weaknesses associating with the embedding UE-method of Chapter 3. In particular, we discretize only in the support of the solution to avoid having extraneous points beyond the solution's free boundary, and also to avoid a necessary reduction in the solution's regularity. Though some progress has been made by the same method, the great challenge remains for a highly irregular solution, e.g. one with very steep/infinite slope near/at the free boundary. For such situations, it is highly difficult to trace the boundary precisely with Darcy's law, since the error in solution is much larger near its free boundary.

In this chapter, we would like to apply a similar numerical method for PME, but in a slightly different direction. Recall that in Chapter 4, the nonembedding method requires us to incorporate

Darcy's law into the original IBVP (1.2), so that it is modified into the following U-formulation

$$\begin{cases} u_t = \nabla \cdot (|u|^m \nabla u)\,, & \text{in} \quad \Omega(t)\,, \quad t \in (t_0, T] \\ u(\boldsymbol{x}, t_0) = u_0(\boldsymbol{x})\,, & \text{in} \quad \Omega(t_0) \\ u(\boldsymbol{x}, t) = 0\,, & \text{on} \quad \Gamma(t) := \partial\Omega(t)\,, \quad t \in (t_0, T]\,, \\ \Gamma'(t) = \lim\limits_{\boldsymbol{x} \to \Gamma(t)^-} -\nabla \left(\dfrac{|u|^m(\boldsymbol{x}, t)}{m} \right) \cdot \hat{\boldsymbol{n}} & \text{in} \quad (t_0, T]\,. \end{cases} \tag{5.1}$$

We also recall that the Barenblatt-Pattle solution for PME (cf. Eqn. (2.14)), by which we measure the accuracy of our method, have very steep/infinite slope near/at its free boundary whenever $m > 1$, and causes very great challenges for our numerical method. Mathematical studies on PME (e.g. see [46]) show that while the PME solution u can have very low regularity over its support, its "Mathematician's pressure" defined by $\frac{|u|^m}{m}$ is very smooth. We would like to find PME solution (e.g. for (5.1)) indirectly via finding a solution for this pressure instead. To this end, let us denote

$$v := \frac{|u|^m}{m} \tag{5.2}$$

and seek to transform the original PME

$$u_t = \nabla \cdot (|u|^m \nabla u) \tag{5.3}$$

to be in terms of v only. From (5.2), we have

$$u = m^{\frac{1}{m}} v^{\frac{1}{m}}\,.$$

Further,

$$\begin{aligned} u_t &= \nabla \cdot (|u|^m \nabla u)\,, \\ \frac{\partial}{\partial t}(m^{\frac{1}{m}} v^{\frac{1}{m}}) &= \nabla \cdot (mv \nabla (m^{\frac{1}{m}} v^{\frac{1}{m}}))\,, \end{aligned}$$

$$
\begin{aligned}
m^{\frac{1}{m}} \frac{1}{m} v^{\frac{1}{m}-1} v_t &= m^{\frac{1}{m}} \nabla \cdot (mv \frac{1}{m} v^{\frac{1}{m}-1} \nabla v)\,, \\
v^{\frac{1}{m}-1} v_t &= m \nabla \cdot (v^{\frac{1}{m}} \nabla v)\,, \\
v^{\frac{1}{m}-1} v_t &= m[\nabla (v^{\frac{1}{m}}) \cdot \nabla v + v^{\frac{1}{m}} \Delta v]\,, \\
v^{\frac{1}{m}-1} v_t &= v^{\frac{1}{m}-1} \nabla v \cdot \nabla v + m v^{\frac{1}{m}} \Delta v\,, \\
v^{\frac{1}{m}-1} v_t &= v^{\frac{1}{m}-1} \nabla v \cdot \nabla v + m v^{\frac{1}{m}-1} v \Delta v\,, \\
v_t &= |\nabla v|^2 + mv \Delta v\,,
\end{aligned}
$$

or

$$
\begin{aligned}
v_t &= mv \Delta v + |\nabla v|^2\,, \\
v_t &= \nabla \cdot (mv \nabla v) - (m-1)|\nabla v|^2\,. \qquad (5.4)
\end{aligned}
$$

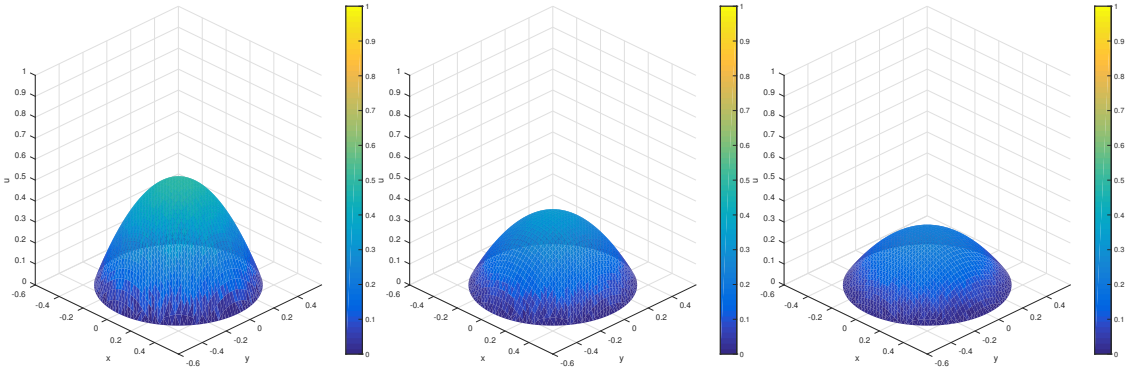
Hence, we can use (5.4) to rewrite the IBVP (5.1) into the "V-formulation" of PME as

$$
\begin{cases}
v_t = \nabla \cdot (mv \nabla v) - (m-1)|\nabla v|^2\,, & \text{in} \quad \Omega(t), t \in (t_0, T] \\
v(\boldsymbol{x}, t_0) = v_0(\boldsymbol{x}) := \frac{(u_0(\boldsymbol{x}))^m}{m}\,, & \text{in} \quad \Omega(t_0) \\
v(\boldsymbol{x}, t) = 0\,, & \text{on} \quad \Gamma(t) := \partial\Omega(t)\,, \quad t \in (t_0, T] \\
\Gamma'(t) = \lim\limits_{\boldsymbol{x} \to \Gamma(t)^-} -\nabla (v(\boldsymbol{x}, t)) \cdot \hat{\boldsymbol{n}}\,, & \text{in} \quad (t_0, T]\,.
\end{cases} \qquad (5.5)
$$

Thus, our moving mesh method applied to this system will be called the "V-method."

System (5.5) also admits a slightly modified form of Barenblatt-Pattle solution (cf. Eqn. (2.14)), and is given by

$$
v(r,t) = \begin{cases}
\frac{1}{m\lambda^{dm}(t)} \left(1 - \left(\frac{r}{r_0 \lambda(t)}\right)^2\right), & \text{for} \quad |r| \le r_0 \lambda(t) \\
0, & \text{for} \quad |r| > r_0 \lambda(t)
\end{cases} \qquad (5.6)
$$



(a) $m = 2$ (b) $m = 3$ (c) $m = 4$

Figure 5.1: The V-formulated Barenblatt-Pattle solutions for PME.

where

$$r = |\boldsymbol{x}|, \quad \lambda(t) = \left(\frac{t}{t_0}\right)^{\frac{1}{2+dm}}, \quad t_0 = \frac{r_0^2 m}{2(2+dm)}. \tag{5.7}$$

As expected, these special V-formulated Barenblatt-Pattle solutions of PME are very smooth on their support for any parameter m. A few examples of these can be seen in the plots of Fig. 5.1.

As in the previous chapters (for the UE- and U-methods), we will rely on these modified special solutions for benchmarking our method, with the L^2-norm error measurement given in (4.2) of Chapter 4. Similar to the previous discussion in Chapter 4, before presenting the main numerical results, we would like to make a parallel (and comparative) investigation on the roles of the parameters and meshing strategies known to be sensitive to the first nonembedding method. Namely, we are interested in the effects of τ, Δt_{max}, and the choice of the metric tensor $\mathbb{M}$ on our method in this situation (see Chapter 4, Pg. 68 and the associating references for a more detailed elaboration on these components). In the following computations, unless stated otherwise, we will assume the final time T in (5.5) is given by $T = (t_0 + 0.1)/2$, where t_0 is given in (5.7), and also that the largest time step to be no greater than 10^{-3} (i.e. $\Delta t_{max} = 10^{-3}$). Additionally, unlike the previous chapter where we let the boundary points be given exactly for these investigations, we will move the boundary points through Darcy's law with a forward-Euler scheme

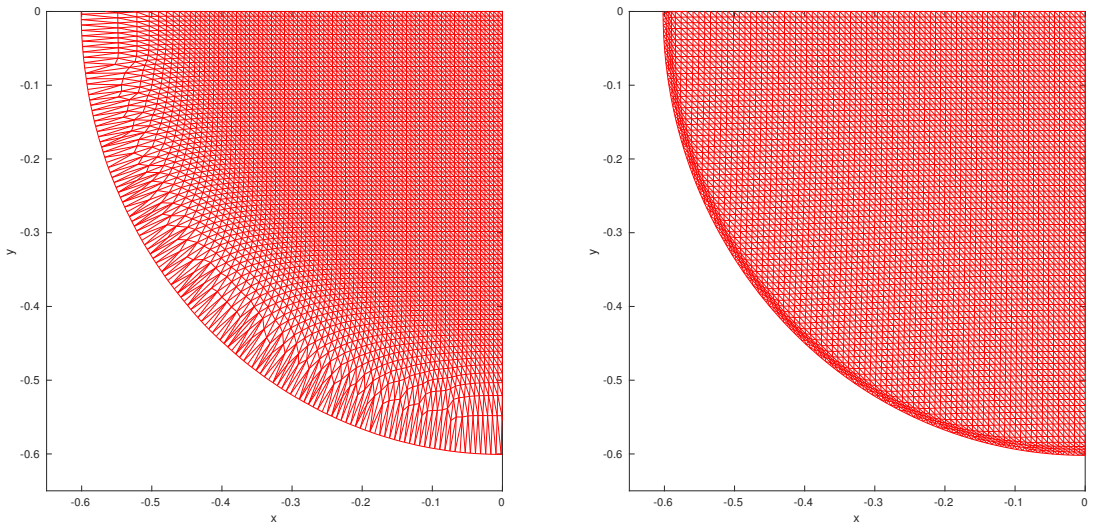
$$\frac{\Gamma_h^{n+1} - \Gamma_h^n}{t_{n+1} - t_n} = -\nabla_h(v) \cdot \hat{\boldsymbol{n}}, \tag{5.8}$$

where $\nabla_h(\cdot)$ represents a scheme for approximating $\lim_{\boldsymbol{x}\to\Gamma_h(t_n)^-}\nabla(\cdot)$ at time $t=t_n$, and $\hat{\boldsymbol{n}}$ represents the unit outward normal to the boundary (See Fig. 2.6).

5.1 Effect of τ

Recall that though the modified Barenblatt-Pattle solutions in (5.6) are more regular within their support than their original counterparts (cf. Eqn (2.14)), and though they do not have steep slopes near/at the edges of their supports, their free boundaries are nevertheless moving. Unless the interior points are made to follow the boundary points sufficiently fast, it will result in the loss of accuracy as we have seen in the previous method (cf. Section 4.1, especially Fig. 4.1). This indeed is the case, as we can see in Fig. 5.2 the representative meshes from two computations with $\tau=10^{-2}$ and $\tau=10^{-4}$. Notice that, similar to the situation of Section 4.1, when the value of τ is not sufficiently small (as to make the mesh points react faster to the change in the metric tensor $\mathbb{M}(\boldsymbol{x},t)$), the interior points could not keep up with the movement of the free boundary (see Fig. 5.2a), and result in a loss of accuracy as indicated by the corresponding convergence history in Fig. 5.3 (cf. Fig. 4.1 for the previous method). On the other hand, a smaller value for τ such as 10^{-4} helps the interior points to follow closely with the free boundary (see Fig. 5.2b), and result in a greater convergence result as seen in the convergence history for the same case in Fig. 5.3. We further notice from the same figure that even for the case of $\tau=10^{-2}$, the convergence order only breaks down for finer meshes. As a practical implementation, from now on, unless otherwise specified, we will use the following adaptive formula for τ (cf. (4.4)) given by

$$\tau=\min\left\{10^{-3},\frac{10^{-1}}{N}\right\}. \tag{5.9}$$



(a) $\tau = 10^{-2}$ (b) $\tau = 10^{-4}$

Figure 5.2: Effect of different values of τ on the outcome of the moving mesh ($m = 2$).

5.2 Effect of Δt_{max}

As we continue to follow the research pattern of the previous chapter, we would like to investigate the role of the largest allowed time step (cf. Eqn. (4.3))

$$\Delta t_{max} = \max_{n=1,\dots,n_f} (t_n - t_{n-1})$$

in our current numerical method. Since the modified Barenblatt-Pattle solutions are very smooth for each $m > 1$, we will only investigate for the case of $m = 3$. We will also use the Hessian-based adaptive meshes, and the choice of τ in (5.9) for these experiments. Fig. 5.4 show the convergence histories for the cases of $\Delta t_{max} = 10^{-2}, 10^{-3}, 10^{-4}$, and 10^{-5}. It turns out that for each of these upper bounds of the time steps, the method performs well and exhibits a second-order convergence. Nevertheless, the convergence histories for cases of $\Delta t_{max} = 10^{-4}$, and 10^{-5} appear to be more stable.

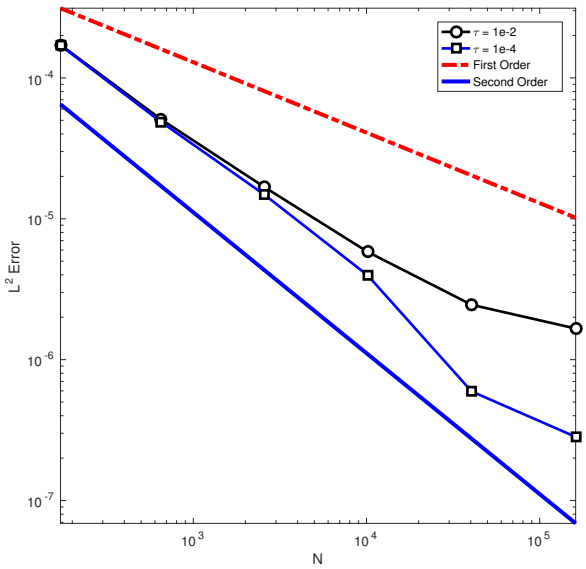


Figure 5.3: Convergence histories for two different choices of τ ($m = 2$).

5.3 Effect of mesh adaptation strategies

Recall from Chapter 4, that the U-method, when applied to the IBVP (4.1) with $m > 1$ and having Barenblatt-Pattle solution, performs differently for different adaptive strategies. In particular, the method performed on adaptive meshes outperforms the same on a uniform mesh. Also between the two adaptive strategies, the Hessian-based adaptivity surpasses the arclength-based one in accuracy and convergence order (cf Section 4.3, and Fig. 4.4b). This great contrast between the mesh adaptation strategies could be accounted for by the peculiar nature of some of these Barenblatt-Pattle solutions, in that they have very steep/infinite slope near/at their free boundaries for $m > 1$. On the other hand, we have also learned from MMPDE theory (e.g. see [26]) that in situations where the PDE solution is sufficiently smooth, no clear advantage might be gained through mesh adaptation, as the mesh remains largely uniform throughout the computation process regardless of the adaptive methods (For example, the Hessian-based adaptive mesh in Fig. 5.5a for the case $m = 3$). In fact, we recall a similar situation in Section 4.3, where the performance between the three mesh adaptation strategies is indistinguishable for the case $m = 1$ of PME, for which the Barenblatt-Pattle solution does not have steep slope at its free boundary.

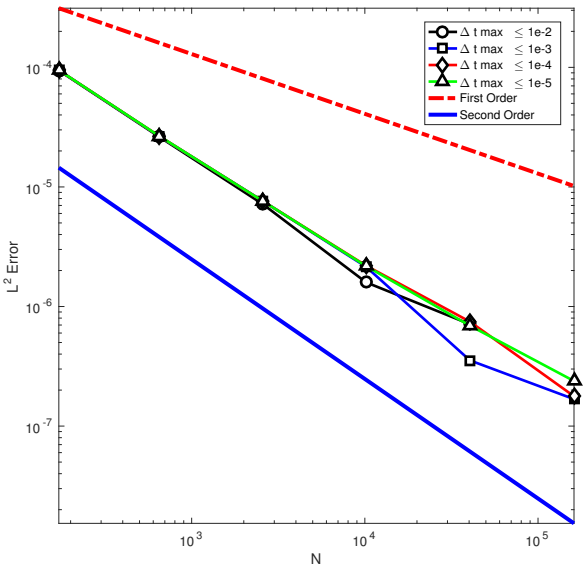


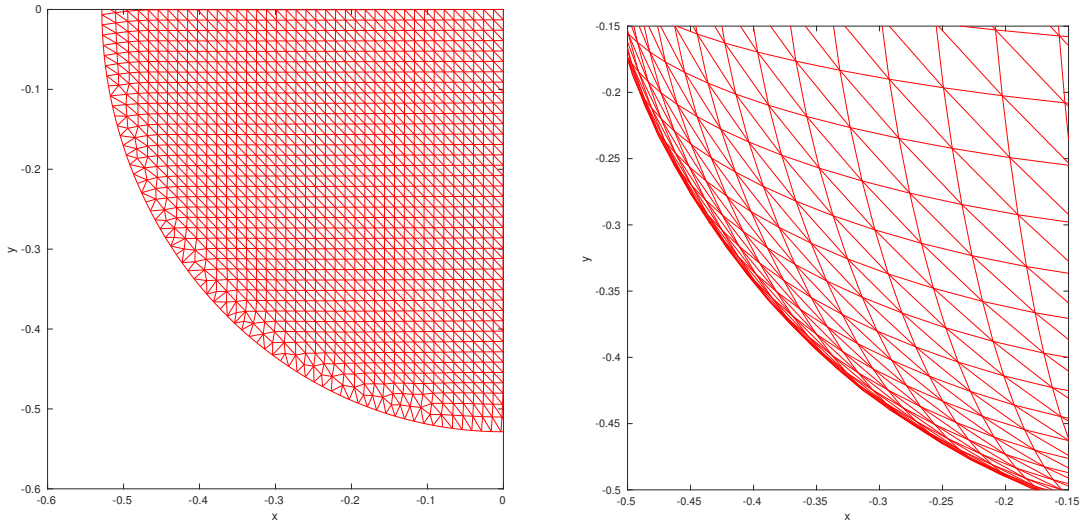
Figure 5.4: Convergence histories for different choices of Δt_{max} ($m = 3$).

For our new V-formulation, the modified Barenblatt-Pattle solutions are very smooth over their supports, including at their free boundaries for $m > 1$. Hence, we do not expect significant differences between these mesh adaptation strategies for our method, when applied to the IBVP (5.5) which have these special solutions. This observation is indeed confirmed through the three convergence histories in Fig. 5.6, which correspond to the three meshing strategies. These numerical experiments have been conducted with $m = 3$, $\Delta t_{max} = 10^{-4}$, and a choice of τ according to (5.9). Ironically, the Hessian-based adaptivity in this case performs slightly worse than the same method applied on the uniform and arclength-based meshes.

Remark 5.3.1. In certain situations where we need to concentrate more mesh points at the boundary, we might use a boundary-based metric tensor introduced in (2.39) such as

$$\mathbb{M}(\boldsymbol{x}) = \frac{1}{U^2 + 10^{-8}}\mathbb{I}, \tag{5.10}$$

by which the MMPDE method clusters the mesh points toward regions where the solution is small, and hence towards the free boundary. A representative mesh based on this tensor is given in Fig. 5.5b, with a corresponding convergence history given in Fig. 5.7b.



(a) Hessian-based adaptive mesh.

(b) Boundary-bashed adaptive mesh.

Figure 5.5: Representative meshes for the Hessian-based and boundary-based mesh adaptations.

5.4 Effect of PME parameter m on the solution

From the previous chapter, we have known that the parameter m of PME in (1.2) (cf. (2.46)) can greatly affect the performance of our moving mesh method. In particular, when the method is applied on uniform meshes, the order of convergence steadily declines as m gets higher, which indeed confirm various error estimations found in the research community. On the other hand, when the method is equipped with Hessian-based adaptivity, the order of convergence is not affected (providing that the boundary points are traced precisely), though the magnitude of the L^2-error gets higher as m increases. Again, since the modified Barenblatt-Pattle solutions in our case are very smooth over their supports, we expect a good convergence order for our method with the new formulation of the IBVP (5.5). Indeed, this is the case, as the convergence histories for the cases $m = 2, 3, 4$ and 5 in Fig. 5.7a show a second-order convergence for all of these. It is worth pointing out that contrary to the previous results for the original formulation (cf. Section 4.4 and Fig. 4.5b), this this situation, the magnitude of the L^2-error decreases as m increases. We also present in Fig. 5.7b a convergence history for the method when applied with the tensor (5.10) given in Remark 5.3.1. We observe that clustering the mesh points toward the boundary in this manner slightly im-

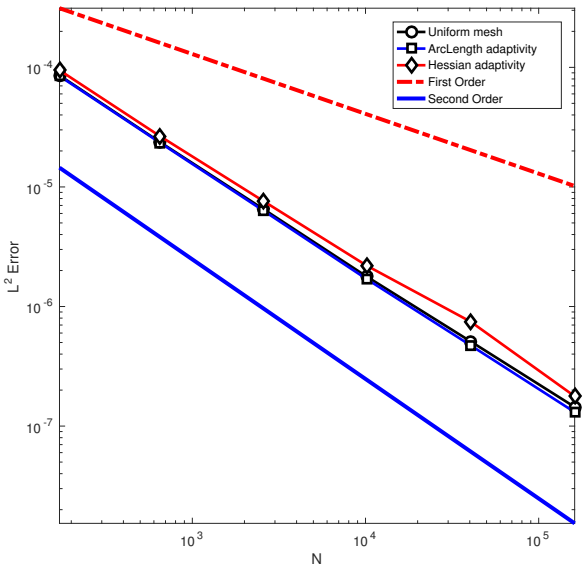


Figure 5.6: Convergence histories for different adaptation strategies.

proves the convergence behavior of the method over fine meshes (comparing with the convergence history for the Hessian-based adaptivity in Fig. 5.7a, where a slight reduction in convergence order is observed for fine meshes).

5.5 Numerical examples

In this section, we present some numerical examples to demonstrate the V-method.

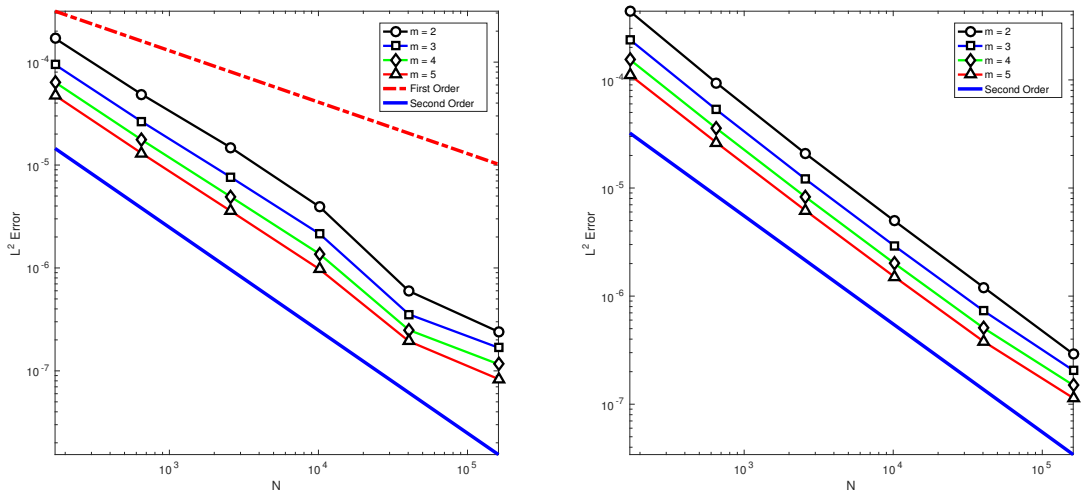
Example 5.5.1 (Barenblatt-Pattle solution). In this example, we apply the V-method to IBVP (5.5), where

$$m = 3\,, \quad r_0 = 0.5\,, \quad \text{and} \quad T \approx 0.07\,.$$

Further, we use a Hessian-based adaptive mesh with $N = 10171$,

$$\tau = \min\left\{10^{-3}, \frac{10^{-1}}{N}\right\}, \quad \text{and} \quad \Delta t_{max} = 10^{-3}\,.$$

In Fig. 5.8, plots of the mesh and its associated solution at the final time are given. Notice that the mesh points are not clustered toward the free boundary, and the solution is not steep at such



(a) Hessian-based adaptive mesh.

(b) Boundary-clustered mesh.

Figure 5.7: Convergence histories for different parameters m for two methods of moving mesh.

location (contrast with Fig. 4.6 of the U-method in Chapter 4). We also consider the results of the same method applied with the metric tensor given in (5.10), where the mesh nodes are deliberately clustered towards the free boundary. Fig. 5.9 shows the mesh and its associated solution given by this boundary-based adaptivity. □

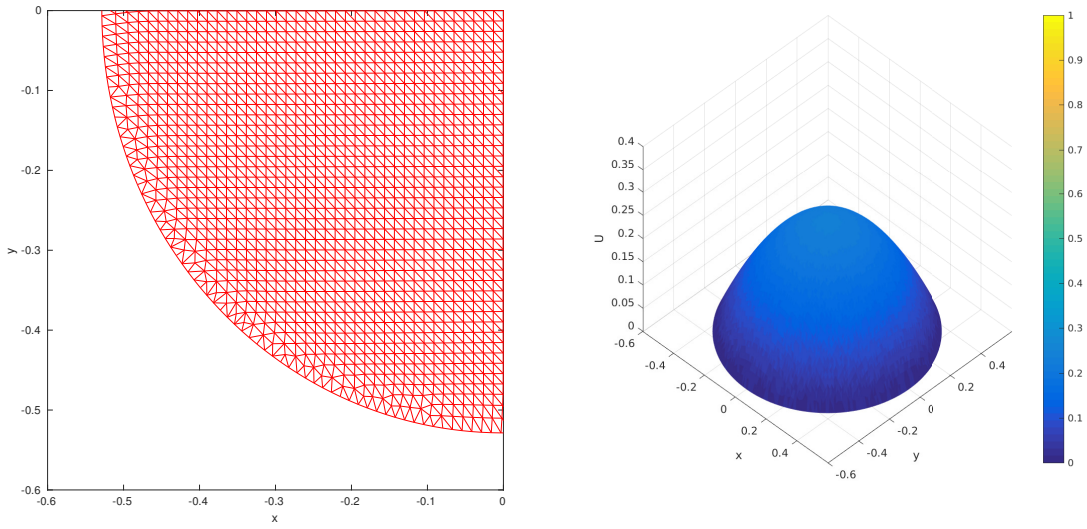
Example 5.5.2 (Waiting time phenomenon). From PME literature, we know that for some special initial solutions, the IBVP (5.5) exhibits the waiting-time phenomenon, where the free boundary does not move initially until a finite amount of time has elapsed. Such is the case for the initial solution

$$v_0(x,y) = \begin{cases} \frac{1}{m}\cos^m\left(\sqrt{x^2+y^2}\right), & \text{for} \quad \sqrt{x^2+y^2} \le \frac{\pi}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

We observe that

$$\nabla\left(\frac{1}{m}\cos^m\left(\sqrt{x^2+y^2}\right)\right) = -\frac{\cos^{m-1}(\sqrt{x^2+y^2})\sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}\begin{bmatrix} x \\ y \end{bmatrix},$$

vanishes at $\sqrt{x^2+y^2} = \frac{\pi}{2}$. Therefore, according to Darcy's law given in (5.5), we should not expect the free boundary to move initially. Here, we experiment our V-method for the case $m = 2$.



(a) Mesh

(b) Computed solution

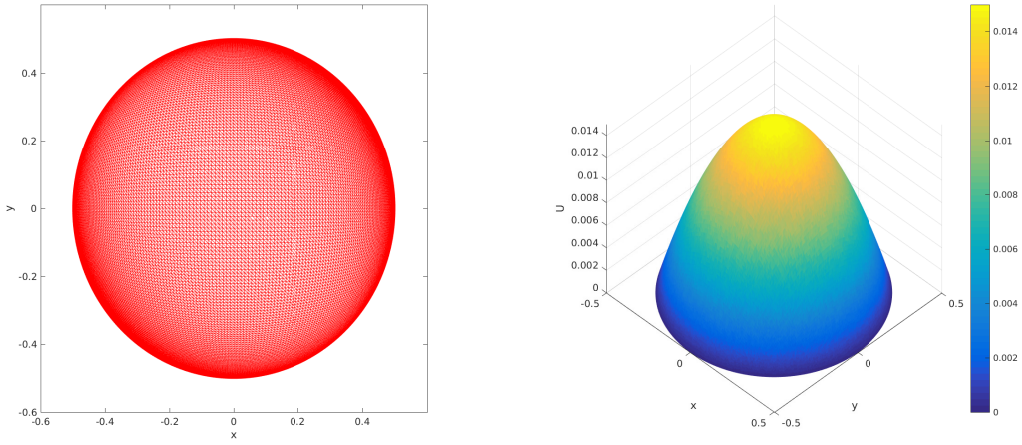
Figure 5.8: Example 5.5.1. The final mesh (close view near (-0.35, -0.35)) and computed solution for $m = 3$ with the Hessian-based mesh adaptation ($N = 10171$).

In Fig. 5.11, we plot a few typical meshes and their associated solutions (with the contrasting circles for representing the initial boundary). To further demonstrate and confirm the waiting-time phenomenon, we also plot the cross section (in the plane $y = 0$) of the solution at various time instants in Fig. 5.12 (where the dashed lines indicate the position of the initial boundary). A closer look suggests that the free boundary does not start moving until around $t = 0.2$. □

Example 5.5.3 (Complex domain). In this example, we apply our V-method to (5.5), with $m = 2$ and

$$v_0(x,y) = \begin{cases} 25\left[0.25^2 - (\sqrt{x^2+y^2} - 0.75)^2\right]^{\frac{3}{2}}, & \text{for} \quad \sqrt{x^2+y^2} \in [0.5,1] \text{ and } (x<0 \text{ or } y<0) \\ 25\left[0.25^2 - x^2 - (y-0.75)^2\right]^{\frac{3}{2}}, & \text{for} \quad x^2+(y-0.75)^2 \le 0.25^2 \text{ and } x \ge 0 \\ 25\left[0.25^2 - (x-0.75)^2 - y^2\right]^{\frac{3}{2}}, & \text{for} \quad (x-0.75)^2+y^2 \le 0.25^2 \text{ and } y \ge 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.12}$$

Here, we would like to cluster more mesh points toward the free boundary, and therefore have used



(a) Mesh.

(b) Computed solution.

Figure 5.9: Example 5.5.1. The final mesh and computed solution for $m = 3$ ($N = 40459$). The mesh nodes are deliberately clustered toward the free boundary.

the boundary-based adaptivity for the MMPDE method with $\mathbb{M}$ given in (2.39), where $\alpha = 10^{-6}$ and $r = \frac{1}{2}$. The partial donut-shaped support pertaining to this initial solution is first seen in the paper of Baines et al. (see [5]). Fig. 5.13 shows the representative meshes and their corresponding solutions. Here, the mesh's evolution is also comparable to the result in [5] (See also Example 4.5.3 in Chapter 4). □

5.6 A discussion on the VE-method

We have seen from Chapter 3 that though the UE-method for PME has some disadvantages in terms of computer memory and solution regularity, it is nonetheless quite accurate (if equipped with a Hessian-based adaptivity) and also very flexible with complicated domains. On the other hand, though the U-method of Chapter 4 and the V-method of this chapter have improved on the weaknesses of the UE-method, they are not as quite robust with more complicated domains. These considerations have encouraged us to explore the "VE-method"–an application of our moving mesh method with embedding approach to the V-formulation of PME, with the hope that the robustness toward complex domains of an embedding method in combination with a high regular-

ity in solution of the V-formulation will produce favorable results. Specifically, we would like to consider the IBVP

$$\begin{cases} v_t = \nabla \cdot (mv\nabla v) - (m-1)|\nabla v|^2\,, & \text{in} \quad \Omega \times (t_0, T] \\ v(\boldsymbol{x}, t_0) = v_0(\boldsymbol{x})\,, & \text{on} \quad \Omega \\ v(\boldsymbol{x}, t) = 0\,, & \text{on} \quad \partial\Omega \times (t_0, T]\,, \end{cases} \tag{5.13}$$

where Ω is sufficiently large to contain the support of $v(\boldsymbol{x}, t)$, for all $t \in (t_0, T]$. Here, by carrying out the VE-method equipped with the Hessian-based adaptivity (similar to that of the UE-method of Chapter 3), and by using the modified Barenblatt-Pattle solution in (5.6) for benchmarking, the method's performance is shown through the convergence histories given in Fig. 5.10 (where the error is measured in the L^2-norm given in (3.2)). We notice that the convergence order is about 1.5, and declines quickly as the mesh gets finer. Currently, we have not yet found an explanation for the drastic decline of performance. More investigations are required for further understanding.

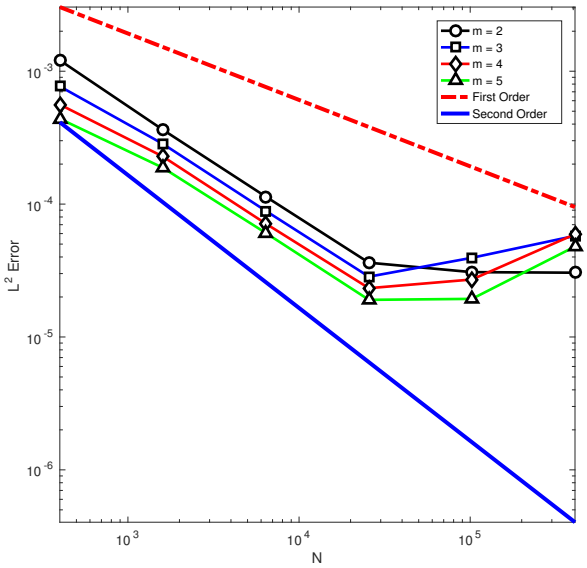
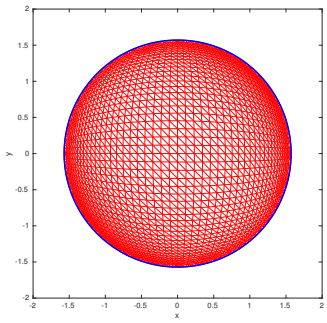
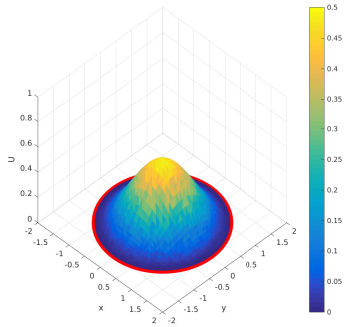


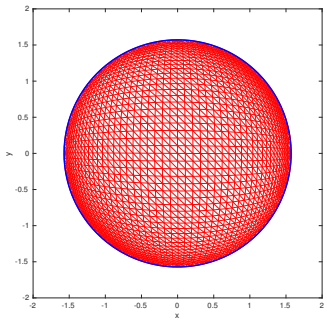
Figure 5.10: The VE-method: Convergence histories for different values of m.



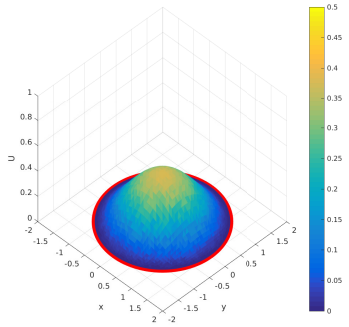
(a) $t = 0.11$



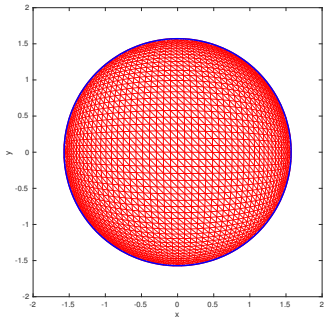
(b) $t = 0.11$



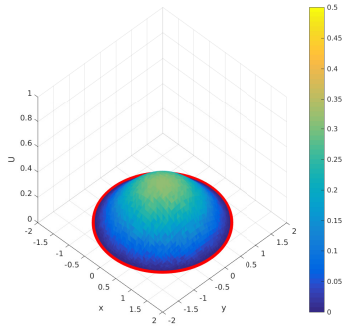
(c) $t = 0.2$



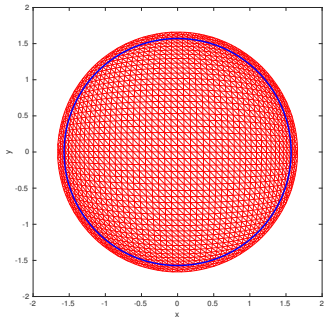
(d) $t = 0.2$



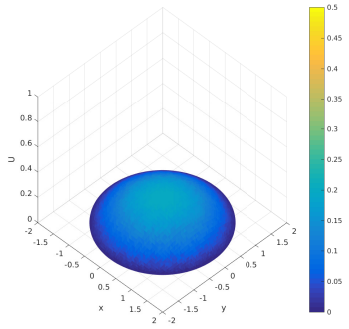
(e) $t = 0.3$



(f) $t = 0.3$

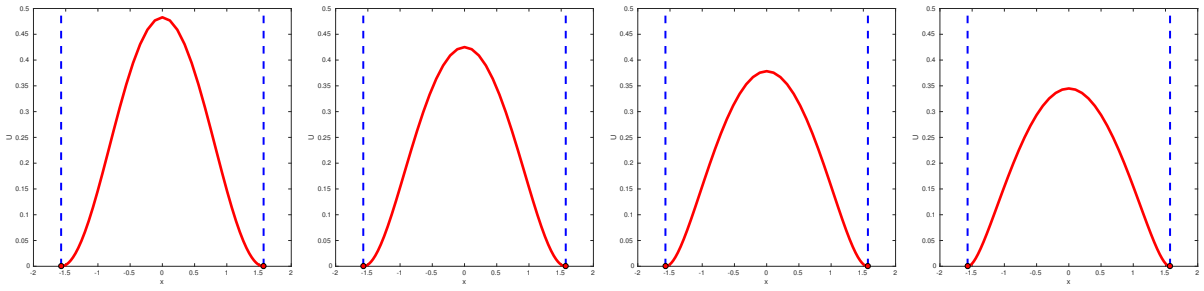


(g) $t = 0.8$

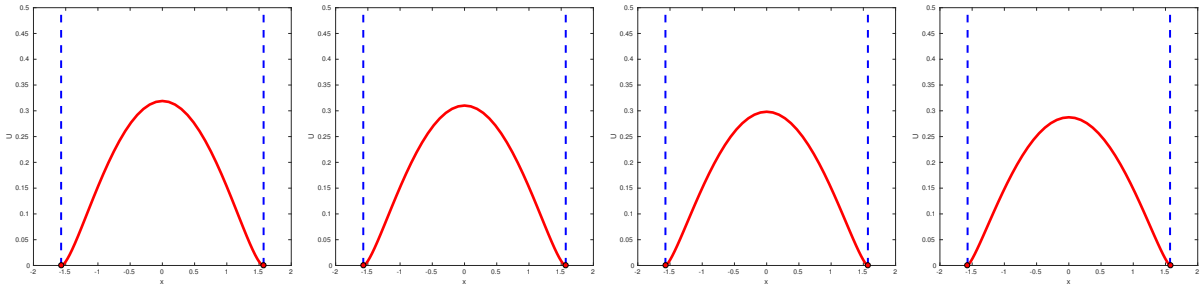


(h) $t = 0.8$

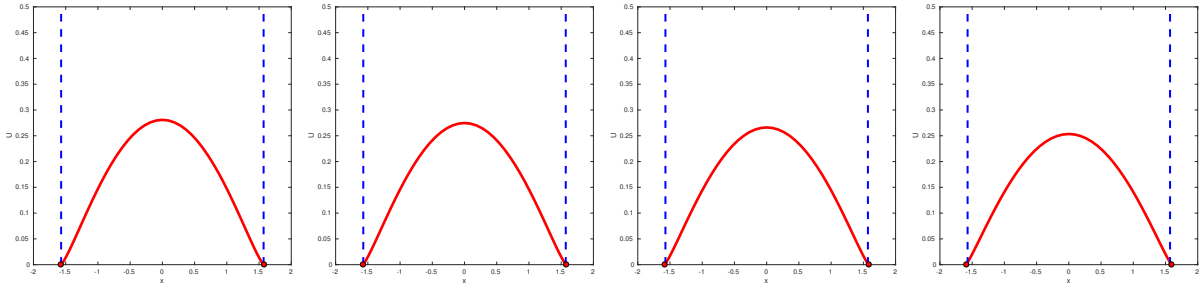
Figure 5.11: Example 5.5.2. A computed solution is shown at various time instants ($N = 4011$).



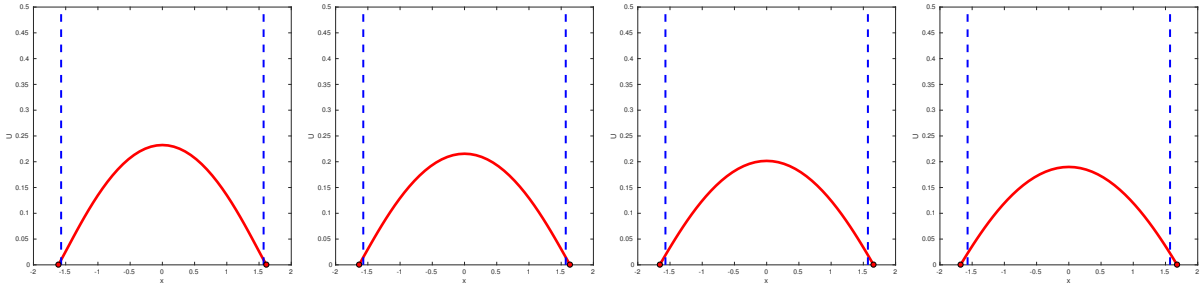
(a) $t = 0.11$ (b) $t = 0.15$ (c) $t = 0.20$ (d) $t = 0.25$



(e) $t = 0.30$ (f) $t = 0.32$ (g) $t = 0.35$ (h) $t = 0.38$

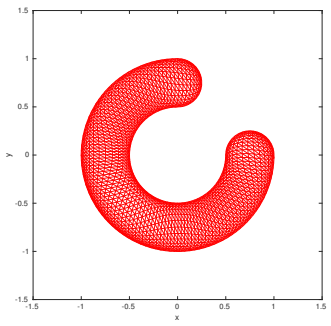


(i) $t = 0.40$ (j) $t = 0.42$ (k) $t = 0.45$ (l) $t = 0.50$

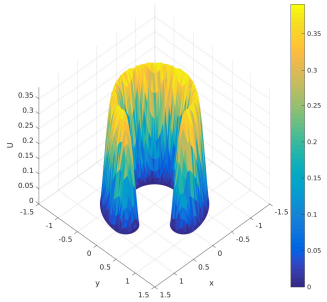


(m) $t = 0.60$ (n) $t = 0.70$ (o) $t = 0.80$ (p) $t = 0.90$

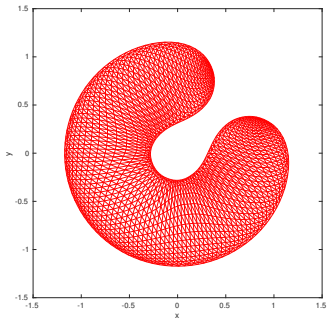
Figure 5.12: Example 5.5.2. The cross section at $y = 0$ of a computed solution is shown at various time instants ($N = 4011$).



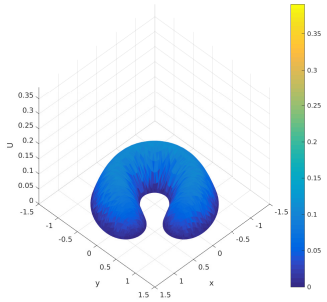
(a) $t = 0.1$



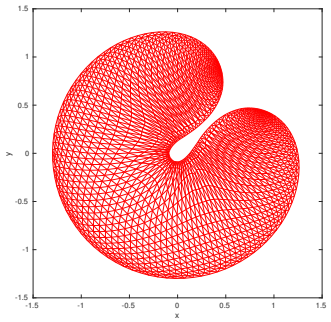
(b) $t = 0.1$



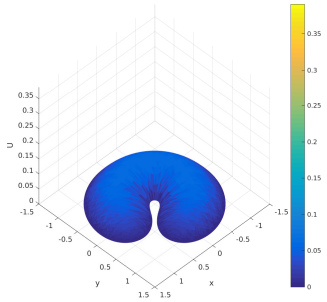
(c) $t = 0.3$



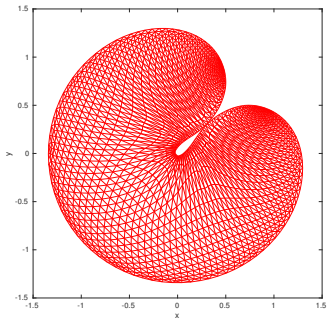
(d) $t = 0.3$



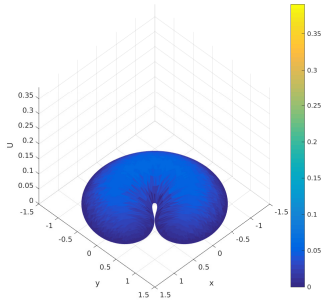
(e) $t = 0.75$



(f) $t = 0.75$



(g) $t = 1$



(h) $t = 1$

Figure 5.13: Example 5.5.3. A computed solution is shown at various time instants ($N = 3242$).

Chapter 6

Conclusion and further remarks

6.1 Conclusion

In this dissertation, we have developed a numerical method for solving numerically the porous medium equation (PME), and in particular, its associated IBVP (1.2). After having understood some peculiar properties associated with such systems, we strongly believe that due to the presence of a free boundary which moves with finite speed, a moving mesh finite element method is potentially more efficient and robust than many others. Moreover, since there are special challenges associated with the PME, most notoriously the lack of regularity of the solution at the free boundary, a moving mesh approach combined with an adaptive method will likely yield more accurate results, with better convergence order. Due to these considerations, we have studied three adaptive moving mesh FE methods, where the mesh generation strategy is based on the MMPDE method of Huang et al. [25, 26].

For the first method–called UE-method, we consider the original IBVP (1.2) and choose a domain sufficiently large as to contain the support of the solution for the duration of the simulation. Hence, for this "immersed-boundary" or "embedding" method, the free boundary itself is part of the solution, and there is no need to trace it explicitly as we have to do for the nonembedding methods of Chapters 4 and 5. The method performs well when applied to the Barenblatt-Pattle

solutions, which produces second-order convergence for the cases $m = 1, 2$, and 3; this surprising performance is in spite of the reduction of the solution's regularity by its extension beyond its support. Though more memory is required for accommodating the mesh points outside the solution's support, the method is quite robust, and capable of handling more complicated structures in the geometry of the solution, and also in the PDE itself. In particular, the simulations of the waiting time (cf. Example 3.2.4), of PME solution with complex support (cf. Examples 3.2.1, 3.2.2, and 3.2.3), and of PME with a variable exponent (i.e. $m = \gamma(\boldsymbol{x})$) and/or absorption term (cf. Example 3.3.2, 3.3.3, and 3.3.4) have yielded a good success.

The second method, presented in Chapter 4, which we have also termed the nonembedding method or U-method, seeks to overcome the two weaknesses of the embedding UE-method (i.e. The necessary reduction of solution's regularity, and the requirement for extraneous points outside solution's support). This method, which has been used commonly by other researchers in the recent years, only discretizes within the support of the PME solution, where the boundary points of the mesh are moved manually via Darcy's law (cf. (2.13)) at each time level. We apply the B-M-P splitting strategy to our discrete system, where we solve separately, yet sequentially, the boundary equation (from the forward-Euler discretization of Darcy's law), the mesh equations (from the MMPDE method), and the physical equations (from the FE discretization of PME on the moving mesh). Such splitting strategy is advantageous since the mesh and physical equations have different structures, and it is prudent for us to deal with these separately. The method has been demonstrated to produce a second-order convergence (cf. Fig. 4.5a), even for PME with parameters $m > 1$, provided that the movement of the boundary is traced precisely. However, if the free boundary is not traced precisely (especially for solutions with steep slope near/at the free boundary), there is a reduction in convergence order of the method over fine meshes (cf. Fig. 4.5b). Therefore, more research on how to improve the precision of boundary movement is much needed. Another shortcoming of the U-method (and also the subsequent V-method) at the present is its lack of robustness for more complicated domains when compared to the UE-method, especially in a situation where two parts of the solution's support come into contact with one another (as in

Examples 3.2.1, 3.2.2, and 3.2.3). Further research is also needed for to overcome this particular issue.

The third method–termed the V-method, discussed in Chapter 5, is essentially the same as the U-method, except that we seek PME solution indirectly by solving for its "Mathematician 's pressure" (5.2) through the modified IBVP (5.5). The modified Barenblatt-Pattle solutions (cf. (5.6)) corresponding to this new setting are very smooth at their free boundaries (see Fig. 5.1). In particular, the gradient of each of these is not steep/infinite at its free boundary, and therefore the boundary movement computation via Darcy's law is more accurate, even with the lower-order forward Euler scheme as in (5.8). The method achieves a second order convergence for all the parameters m considered in our study (cf. Fig. 5.7a), with the L^2-error magnitude decreases as m increases. Furthermore, in applications where the mesh nodes are preferred to be clustered toward the boundary, we may use the special boundary-based metric tensor given in (5.10) with the MMPDE method for the purpose.

All together, these three methods show an advancement in the state of the art for the numerical solution of PME.

6.2 Future Research

The following topics are interesting and quite needed for improving the progress of numerical solution for PME and other free boundary problems.

- **Boundary movement**: As we have noticed in Section 4.5, the accuracy of the nonembedding U-method, especially when applied to PME solution with low regularity at the boundary, is highly dependent on the precision of the boundary movement governed by Darcy's law (cf. Fig. 4.5). In particular, for such situations, a lower-order approximation of the free boundary leads to a loss of accuracy for our method over very fine meshes. Therefore, we need to develop a better method to trace the boundary points more precisely.

- **Degenerate mesh**: For the U-method in Chapter 4, we know that if the solution has very

steep gradient at the free boundary, the Hessian-based MMPDE adaptation will push the mesh points toward the same location. While this behavior greatly improves the accuracy and even the convergence order for our method, it also has its weaknesses. As we have seen in Fig. 4.6b, when the mesh points move too close to the boundary, the mesh elements near/at the same boundary tend to be very flat, making the mesh become anisotropic (and almost degenerate). As a consequence, it is more challenging to recover the Hessian of the solution, making mesh generation more difficult. Hence, more research is needed to overcome this shortcoming.

- **Higher dimensions**: We have not yet seen a PME result in a higher-dimensional setting. For a future project, we would like to study the numerical solution of PME in three or higher dimensions.

- **Other nonlinear problems**: Though much progress has been made for a numerical method for solving PME, this PDE is still a simple example of other more general nonlinear and/or free boundary problems. We would like to explore and improve our method towards being more capable for handling such problems.

- **The VE-formulation**: Currently, we have some difficulties in applying the embedding method to the V-formulation (cf. Section 5.6). In particular, our VE-method breaks down for very fine meshes. Further research is required for understanding and overcoming the issue.

Publications

1. Cuong Ngo and Weizhang Huang. Monotone finite difference schemes for anisotropic diffusion problems via nonnegative directional splittings. *Communications in Computational Physics*, 19(2):473–495, 2016. (DOI: 10.4208/cicp.280315.140815a) (arXiv:1503.08177)

2. Cuong Ngo and Weizhang Huang. A Study on Moving Mesh Finite Element Solution of the Porous Medium Equation. *Journal of Computational Physics*, 331:357–380, 2017. (DOI: 10.1016/j.jcp.2016.11.045) (arXiv:1605.03570)

Bibliography

[1] S. N. Antontsev and S. I. Shmarev. A model porous medium equation with variable exponent of nonlinearity: existence, uniqueness and localization properties of solutions. *Nonlinear Anal.*, 60(3):515–545, 2005.

[2] D. G. Aronson. Regularity properties of flows through porous media. *SIAM J. Appl. Math.*, 17:461–467, 1969.

[3] M. J. Baines. *Moving Finite Elements*. Oxford University Press, Oxford, 1994.

[4] M. J. Baines, M. E. Hubbard, and P. K. Jimack. A moving mesh finite element algorithm for fluid flow problems with moving boundaries. *I. J. Numer. Meth. Fluids*, 47:1077–1083, 2005. 8th ICFD Conference on Numerical Methods for Fluid Dynamics. Part 2.

[5] M. J. Baines, M. E. Hubbard, and P. K. Jimack. A moving mesh finite element algorithm for the adaptive solution of time-dependent partial differential equations with moving boundaries. *Appl. Numer. Math.*, 54:450–469, 2005.

[6] M. J. Baines, M. E. Hubbard, and P. K. Jimack. Velocity-based moving mesh methods for nonlinear partial differential equations. *Commun. Comput. Phys.*, 10:509–576, 2011.

[7] M. J. Baines, M. E. Hubbard, P. K. Jimack, and A. C. Jones. Scale-invariant moving finite elements for nonlinear partial differential equations in two dimensions. *Appl. Numer. Math.*, 56:230–252, 2006.

[8] W. Bo and M. Shashkov. R-adaptive reconnection-based arbitrary Lagrangian Eulerian method-R-ReALE. *J. Math. Study*, 48:125–167, 2015.

[9] C. Budd, G. Collins, W. Huang, and R. D. Russell. Self-similar numerical solutions of the porous medium equation using moving mesh methods. *Phil. Trans. R. Soc. Lond. A*, 357:1047–1078, 1999.

[10] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.

[11] W. Cao, W. Huang, and R. D. Russell. A moving mesh method based on the geometric conservation law. *SIAM J. Sci. Comput.*, 24:118–142, 2002.

[12] P. Daskalopoulos and R. Hamilton. Regularity of the free boundary for the porous medium equation. *J. Amer. Math. Soc.*, 11:899–965, 1998.

[13] J. C. M. Duque, R. M. P. Almeida, and S. N. Antontsev. Convergence of the finite element method for the porous media equation with variable exponent. *SIAM J. Numer. Anal.*, 51(6):3483–3504, 2013.

[14] J. C. M. Duque, R. M. P. Almeida, and S. N. Antontsev. Numerical study of the porous medium equation with absorption, variable exponents of nonlinearity and free boundary. *Appl. Math. Comput.*, 235:137–147, 2014.

[15] J. C. M. Duque, R. M. P. Almeida, and S. N. Antontsev. Application of the moving mesh method to the porous medium equation with variable exponent. *Math. Comput. Simulation*, 118:177–185, 2015.

[16] C. Ebmeyer. Error estimates for a class of degenerate parabolic equations. *SIAM J. Numer. Anal.*, 35:1095–1112, 1998.

[17] C. Ebmeyer and W. B. Liu. Finite element approximation of the fast diffusion and the porous medium equations. *SIAM J. Numer. Anal.*, 46:2393–2410, 2008.

[18] E. Emmrich and D. Šiška. Full discretization of the porous medium/fast diffusion equation based on its very weak formulation. *Commun. Math. Sci.*, 10:1055–1080, 2012.

[19] S. González-Pinto, J. I. Montijano, and S. Pérez-Rodríguez. Two-step error estimators for implicit Runge-Kutta methods applied to stiff systems. *ACM Trans. Math. Software*, 30(1):1–18, 2004.

[20] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996. Stiff and differential-algebraic problems.

[21] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14:227–253, 1974.

[22] W. Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174:903–924, 2001.

[23] W. Huang. Mathematical principles of anisotropic mesh adaptation. *Comm. Comput. Phys.*, 1:276–310, 2006.

[24] W. Huang and L. Kamenski. A geometric discretization and a simple implementation for variational mesh generation and adaptation. *J. Comput. Phys.*, 301:322–337, 2015. (arXiv:1410.7872).

[25] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle. *SIAM J. Numer. Anal.*, 31:709–730, 1994.

[26] W. Huang and R. D. Russell. *Adaptive Moving Mesh Methods*. Springer, New York, 2011. Applied Mathematical Sciences Series, Vol. 174.

[27] W. Huang and W. Sun. Variational mesh adaptation II: error estimates and monitor functions. *J. Comput. Phys.*, 184:619–648, 2003.

[28] P. K. Jimack and A. J. Wathen. Temporal derivatives in the finite-element method on continuously deforming grids. *SIAM J. Numer. Anal.*, 28:990–1003, 1991.

[29] A. S. Kalašnikov. Formation of singularities in solutions of the equation of nonstationary filtration. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 7:440–444, 1967.

[30] L. Kamenski. *Anisotropic Mesh Adaptation Based on Hessian Recovery and A Posteriori Error Estimates*. PhD thesis, TU Darmstadt, 2009.

[31] L. Kamenski and W. Huang. How a nonconvergent recovered Hessian works in mesh adaptation. *SIAM J. Numer. Anal.*, 52:1692–1708, 2014. (arXiv:1211.2877).

[32] B. F. Knerr. The behavior of the support of solutions of the equation of nonlinear heat conduction with absorption in one dimension. *Trans. Amer. Math. Soc.*, 249:409–424, 1979.

[33] R. Li, T. Tang, and P. W. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys.*, 170:562–588, 2001.

[34] S. Lian, W. Gao, C. Cao, and H. Yuan. Study of the solutions to a model porous medium equation with variable exponent of nonlinearity. *J. Math. Anal. Appl.*, 342:27–38, 2008.

[35] G. J. Liao and D. Anderson. A new approach to grid generation. *Appl. Anal.*, 44:285–298, 1992.

[36] K. Miller and R. Miller. Moving finite elements I. *SIAM J. Numer. Anal.*, 18:1019–1032, 1981.

[37] C. Ngo and W. Huang. Monotone finite difference schemes for anisotropic diffusion problems via nonnegative directional splittings. *Comm. Comput. Phys.*, 19:473–495, 2016.

[38] R. H. Nochetto and C. Verdi. Approximation of degenerate parabolic problems using numerical integration. *SIAM J. Numer. Anal.*, 25:784–814, 1988.

[39] O. A. Oleĭnik, A. S. Kalašinkov, and Y. Čžou. The Cauchy problem and boundary problems for equations of the type of non-stationary filtration. *Izv. Akad. Nauk SSSR. Ser. Mat.*, 22:667–704, 1958.

[40] M. E. Rose. Numerical methods for flows through porous media. I. *Math. Comp.*, 40:435–467, 1983.

[41] J. Rulla and N. J. Walkington. Optimal rates of convergence for degenerate parabolic problems in two dimensions. *SIAM J. Numer. Anal.*, 33:56–67, 1996.

[42] S. Shmarev. Interfaces in solutions of diffusion-absorption equations in arbitrary space dimension. In *Trends in partial differential equations of mathematical physics*, volume 61 of *Progr. Nonlinear Differential Equations Appl.*, pages 257–273. Birkhäuser, Basel, 2005.

[43] S. I. Shmarev. Interfaces in multidimensional diffusion equations with absorption terms. *Nonlinear Anal.*, 53:791–828, 2003.

[44] E. A. Socolovsky. On numerical methods for degenerate parabolic problems. *PhD Thesis. Carnegie Mellon University*, 1984.

[45] T. Tang. Moving mesh methods for computational fluid dynamics flow and transport. In *Recent Advances in Adaptive Computation (Hangzhou, 2004)*, volume 383 of *AMS Contemporary Mathematics*, pages 141–173. Amer. Math. Soc., Providence, RI, 2005.

[46] J. L. Vázquez. *The porous medium equation*. Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, Oxford, 2007. Mathematical theory.

[47] D. Wei and L. Lefton. A priori L^ρ error estimates for Galerkin approximations to porous medium and fast diffusion equations. *Math. Comp.*, 68:971–989, 1999.

[48] X. Xu, W. Huang, R. D. Russell, and J. F. Williams. Convergence of de Boor's algorithm for the generation of equidistributing meshes. *IMA J. Numer. Anal.* 31(2):580–596, 2011.

[49] Q. Zhang and Z.-L. Wu. Numerical simulation for porous medium equation by local discontinuous Galerkin finite element method. *J. Sci. Comput.*, 38:127–148, 2009.